R-1258-ARPA

April 1974

A Prototype Computer Program for Interactive Computer-Administered Admissible Probability Measurement

William L. Sibley

A Report prepared for

DEFENSE ADVANCED RESEARCH PROJECTS AGENCY

Rand

SANTA MONICA, CA. 90406

R-1258-ARPA
April 1974

A Prototype Computer Program for Interactive Computer-Administered Admissible Probability Measurement

William L. Sibley

A Report prepared for

DEFENSE ADVANCED RESEARCH PROJECTS AGENCY

Rand
SANTA MONICA, CA. 90406

PREFACE

This report was prepared as part of Rand's OSD Manpower, Personnel, and Training Program, sponsored by the Human Resources Research Office of the Defense Advanced Research Projects Agency (ARPA). With manpower considerations assuming an ever-increasing importance for the Department of Defense, the purpose of this research program is to bring new methodologies to bear on present and future military manpower problems.

The report describes a prototype computer program based on "admissible probability measurement"--a testing procedure which allows a user to express his uncertainty, in terms of probabilities, about the correctness of his answers to multiple-choice questions. A logarithmic scoring system encourages the user to "reproduce" or "reveal" his personal, subjective probability of correctness for each of the choices. The on-line, graphical nature of the computer program provides an excellent environment for allowing the user to interact with such a scoring system.

Admissible probability measurement must be considered carefully in terms of the environment in which it is administered and its possible effects upon users. These issues, among others, are discussed in a companion report, R-1371-ARPA, *A Rationale and Some Implications of Computer-Administered Admissible Probability Measurement* by Emir Shuford, Jr. and Thomas A. Brown (forthcoming).

This report should be useful to administrators concerned with advanced testing methods, to instructors of courses involving the estimation of probabilities, and to computer programmers who wish to implement a version of computer-administered admissible probability measurement.

SUMMARY

This report describes an experimental computer implementation of admissible probability measurement techniques. These measurement techniques rely on a scoring system that encourages the user to "reproduce" or "reveal" his own estimate of the probability of correctness of an answer to a multiple-choice question.

A "track record" (in the form of an external validity graph) of the user's probability estimating performance provides the means for estimating how much information he has and how realistically that level of knowledge is assessed.

The computer system allows the user to explore the relationships between his assignment of probability of correctness to answers and the possible score outcome. In addition, the computer analyzes the external validity graph generated during test administration to provide feedback about his present state of knowledge and how realistically that knowledge is applied. The ease of use of the computer system and the speed of analysis permit cycling through tests quickly, thus removing a major barrier from the task of learning to make better assessments of the user's knowledge.

ACKNOWLEDGMENTS

R-1258

CONTENTS

FIGURES

I. INTRODUCTION

PURPOSE OF THE REPORT

It is the purpose of this report to document a collection of computer procedures for administering admissible probability measurement (APM) techniques. The report is divided into the following parts:

- An introduction to the motivations and mathematical foundations behind APM.
- A description of the environment (both hardware and software) in which the computer procedures were developed, and a description of the operation of the computer procedures.
- A short discussion of planned modifications and extensions of the present system.
- A brief introduction to the PLATO IV system at the University of Illinois. (A version of APM is planned for the PLATO IV system.)
- Two appendices containing detailed descriptions of support software as well as listings of the computer procedures.

This report is sufficiently complete to allow an interested investigator to develop his own operating version of computer-administered APM.

MOTIVATIONS AND FOUNDATIONS

The prime motivation for APM is simple and straightforward: We need to measure more accurately the "state of knowledge" of students in a large variety of subject areas. An improved measuring technique would be of considerable value not only in testing a student but also in evaluating and planning curricula.[1,2] As indicated by Gardner,[2] planning is especially critical for those courses of instruction that are individually tailored to the student.

A second but still important motivation is that the estimation of probabilities is a critical skill in some endeavors, such as weather prediction and intelligence analysis. APM is especially applicable in the training of students to use effectively all of their facts and reasons in the estimation of probabilities.

A corollary of the need to train specialists in probability estimation is the idea that, in fact, we are all engaged in probability estimation. That is, there is a need for us all to convey accurately and effectively our opinions on the possibilities of various outcomes. If that is the case, we must learn to speak the language of probabilities precisely and with a common understanding.

So what is admissible probability measurement and how is it applied? APM will be discussed in terms of one specific application even though it has greater generality than may be apparent from this discussion.

Consider the multiple-choice test item below.

QUESTION: The 11*th* president of the United States was:

        1. George Washington
        2. James Polk
        3. Franklin Pierce

We are generally required to respond to such a question with what amounts to a probability distribution, as illustrated below.

QUESTION: The 11*th* president of the United States was:

$$\begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$$

    1. George Washington
    2. James Polk
    3. Franklin Pierce

Probability
Distribution

However, a more general tendency would be to respond with the probability distribution shown here.

> QUESTION: The 11*th* president of the United States was:
>
> $$\begin{bmatrix} 0 \\ 1/2 \\ 1/2 \end{bmatrix}$$
>
> 1. George Washington
> 2. James Polk
> 3. Franklin Pierce
>
> Revised
> Probability
> Distribution

In fact, if confronted with the following question, a response might very well be, "I can't distinguish among the alternatives." That is, "as far as I can tell from my facts and reasons, each alternative is equally likely."

QUESTION: The 11*th* president of the United States was:

$$\begin{bmatrix} 1/3 \\ 1/3 \\ 1/3 \end{bmatrix}$$

1. John Tyler
2. James Polk
3. Franklin Pierce

It is evident that requiring responses of 0 or 1 can mask a large variety of states of knowledge and can introduce guessing on the part of the student.

However, if greater freedom of response is allowed, there remains the problem of how to score such a response. That is precisely the function of the "admissible," or "proper," or "reproducing" scoring system.[3] Such a scoring system encourages the strategy of responding with exactly what is perceived to be the probability distribution, based on the user's own facts and reasons.

ADMISSIBLE SCORING SYSTEMS

The admissible scoring system can be explained in terms of a gambling example (a more complete discussion can be found in Ref. 3). For the sake of simplicity, suppose that a student were faced with a true-false question. In that situation he may be viewed as a gambler faced with determining the probability that the correct response is "true." Suppose also that there were a gambling house in which there

were $\phi(u)du$ wagers available at odds $(1 - u)/u$ (the "correct odds" for an event of probability u). If the student wished to maximize his expected return and believed that the probability of the event taking place were p, he would take all wagers at odds better than $(1 - p)/p$. Similarly, he would take all wagers on the event *not* taking place at odds better than those appropriate for probability $1 - p$.

The mathematics of admissible scoring systems are illustrated below. Equation (1) corresponds to our true-or-false discussion. Equation (2) extends those notions to an n-alternative multiple-choice question. Equation (3) adjusts the equations to give a score of zero for the "unable to distinguish" response of $p_i = \frac{1}{n}$ for $i = 1,n$. What these integrals represent is the net settlement made between the gambling house and the student, after the correct alternative has been revealed. It is the sum of settlements on an infinite number of infinitesimal wagers, each of which appeared to the student to be favorable to him.

True or false

$$\text{payoff (if "true" occurs)} = \int_0^p \phi(u) \frac{1 - u}{u} du - \int_0^p \phi(u)du \qquad (1)$$

Multiple choice (n alternatives)

$$\begin{array}{c}\text{payoff} \\ \text{if } ith \text{ event occurs)}\end{array} = \int_0^{p_i} \frac{\phi(u)du}{u} - \sum_{j=1}^n \int_0^{p_j} \phi(u)du \qquad (2)$$

Adjusted multiple choice

$$\begin{array}{c}\text{payoff} \\ \text{(if } ith \text{ event occurs)}\end{array} = \int_{1/n}^{p_i} \frac{\phi(u)du}{u} - \sum_{j=1}^n \int_{1/n}^{p_j} \phi(u)du \qquad (3)$$

Equations (4) and (5) below illustrate the logarithmic admissible scoring system which has the property of depending only on the probability ascribed to each alternative independent of the probabilities ascribed to the remaining alternatives.

The logarithmic scoring system $\phi(u) = 1$:

$$\text{payoff (if } i\textit{th} \text{ event occurs)} = \int_{1/n}^{p_i} \frac{du}{u} - \sum_{j=1}^{n} \int_{1/n}^{p_j} du$$

$$= \log(p_i) - \log(\tfrac{1}{n}) - \sum_{j=1}^{n} (p_j - \tfrac{1}{n})$$

$$= \log(np_i) \text{ (assuming } \sum_{j=1}^{n} p_j = 1) \qquad (4)$$

An adjusted system for three alternatives and a score range of about 100 points:

$$= 50 \log(3p_i) \quad 0.01 \leq p_i \leq 0.98 \qquad (5)$$

Equation (5) illustrates the particular version of the logarithmic system we use in APM. To avoid infinitely large penalties, no probability less than 0.01 may be assigned.

The gambling illustration shows that it is in the student's best interest to gauge his probabilities accurately and thus to make no unnecessary bets. That is, the scoring system does encourage him to reproduce his own probabilities accurately.

However, how does the student learn to "play the game" and improve the way he brings his facts and reasons to bear on the determination of the probability he ascribes to each alternative? As Gardner points out, "It seems likely that the more exposure the students have to [probability] tests, the more realistically they will evaluate test items and

their own knowledge, thus providing a more valid indication as to their actual level of achievement."[2] We believe the answer to the question lies in the computer system that will be described next.

THE ROLE OF THE COMPUTER

As can be seen in the figures that follow, the computer-based system we have developed requires a computer terminal with graphics capabilities and some means of allowing the student to interact with it in elapsed times of at most a few seconds. The system was developed on a highly interactive console connected to a powerful IBM 360/65 computer.

Figure 1 illustrates the display the student sees while answering a question. He assigns a probability to each alternative by choosing a point in the interior or on the boundary of the equilateral triangle. In such a triangle, the sum of the distances of the point from each side of the triangle is a constant and can be scaled to 1. Thus each point determines a probability distribution for the three alternatives. The probabilities are therefore naturally constrained to sum to 1. The logarithmic scores corresponding to each alternative appear as both numeric scores and columns of dots. The labels A, B, and C create the correspondences between the alternatives, the triangle vertices, and the scores. The student may choose different points within the triangle until he is satisfied with his score distribution. He may then select the QUESTION ANSWER option and the terminal displays his cumulative score as illustrated by Fig. 2.

The cumulative score is presented in the form of the maximum (or minimum) score attainable, question by question (Fig. 2). A probability of 0.98 assigned to a correct answer would result in a gain of 24 points, and 0.98 assigned to an incorrect alternative would result in a loss of 76 points. Thus, the student has question-by-question feedback about the relationship of his probability assignments to his score and may adjust his actions accordingly. That is, he may develop a "feel" for the scoring system. Under usual test administration procedures, the delay from testing to reporting the scores is so long that it prevents the development of a "feel" for the scoring system. Moreover, the scores are usually not reported on an item by item basis. This portion

20. PERSPICACITY

A   B   C

A: SUSPICION

B: ACUTE MENTAL VISION

C: PERSISTENCE

-18   16   -17
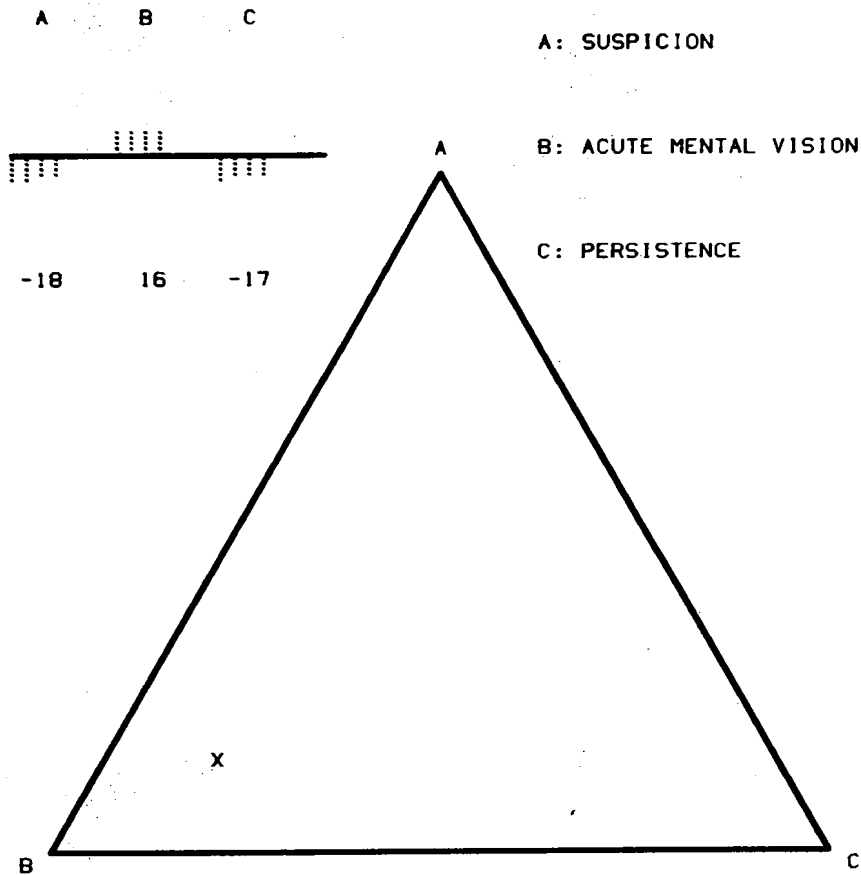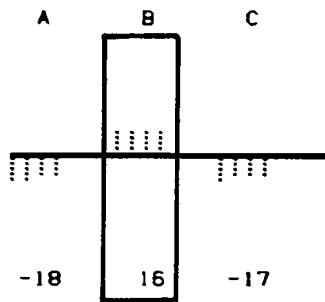
SCORE
PAGE

QUESTION
ANSWER

PREVIOUS
QUESTION

NEXT
QUESTION

Fig. 1 — Display for a question

20. PERSPICACITY
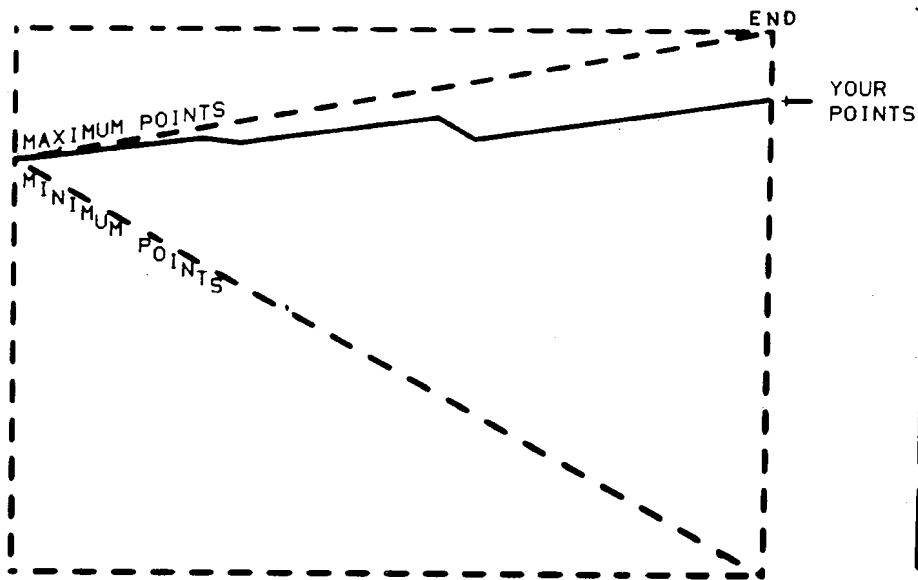

A: SUSPICION

B: ACUTE MENTAL VISION

C: PERSISTENCE

A   B   C

-18   16   -17

Fig. 2 — Cumulative score for a question

of the computer-based system gives the student a maximum amount of experience with the scoring system and its immediate consequences.

Even though the student may learn to express his own probabilities accurately, the question remains as to how those probabilities relate to reality. That question can be answered in part by his external validity graph. [3]

THE EXTERNAL VALIDITY GRAPH

The external validity graph can be used to determine how accurately the student assesses the predictive validity of his facts and reasons. It can be thought of as his "track record" as an assessor. We may collect all those events that he claims have a probability of, say, 0.80 of occurring and determine what percentage of them did in fact occur. If he were a perfect assessor, we would expect roughly 80 percent of them to occur. Figure 3 illustrates such a graph when the number of observations is unlimited. The diagonal AB represents a perfect assessor. If we collect, say for 20 questions, all the probabilities used by a student and compute the relative frequency with which each particular probability is associated with a correct alternative, we may construct an external validity graph for that student taking that test. We then assume that the relationship between student response and relative frequency can be approximated satisfactorily by a straight line simple linear regression.

We call the resulting line the student's realism line. The lines A'B' and A"B" in Fig. 3 illustrate possible results. The line A'B' has the following interpretation: About 60 percent of the events that were given a probability of 1.0 of occurring did in fact occur. We interpret this to mean that the student tends to overvalue his facts and reasons. Similarly, interpretation of the line A"B" shows that the events that were given a probability of 0.60 (or greater) of occurring did in fact occur 100 percent of the time. In other words, the student tends to undervalue his facts and reasons. We may further interpret the realism line in the following way: "When he responded x he should have responded with $Ax + B$," where $Ax + B$ is the equation of his realism line. We may then transform the student's probabilities,

R-1258
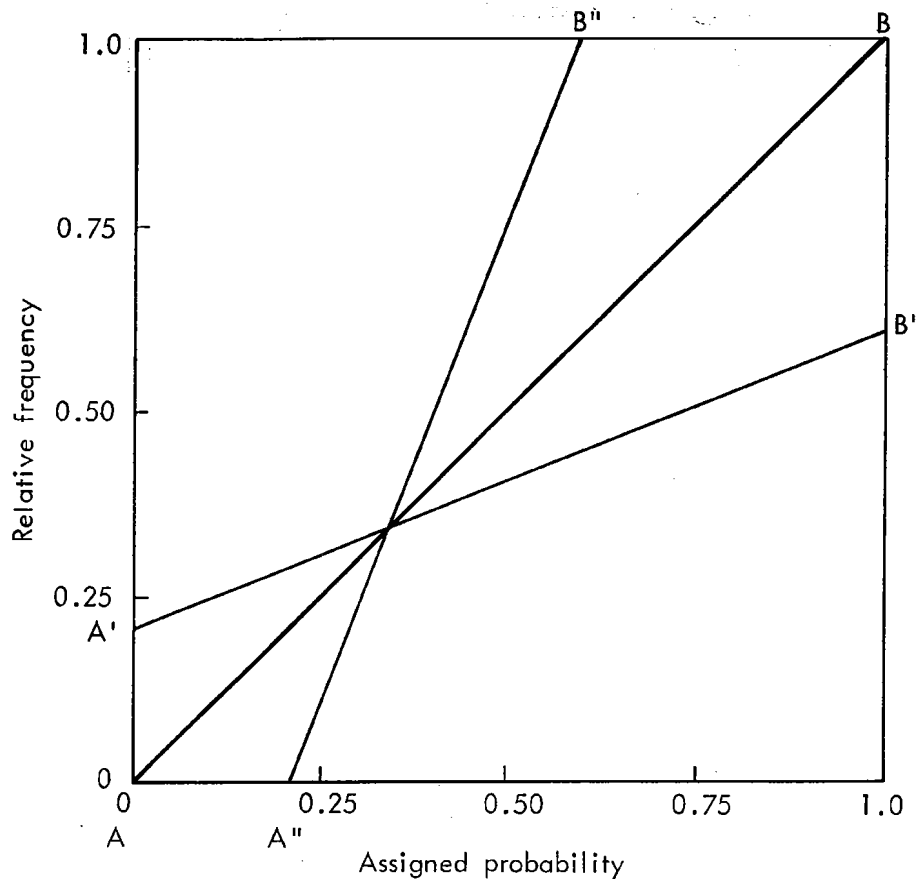
Fig. 3 — External validity graph and realism lines

using his realism line, and recalculate his score on the basis of the
new probability assignments. This revised score can then be used to
decompose his original score into the portion attributable to bias in
his assignment of probabilities and the portion attributable to lack
of information about the subject matter itself. The lower section of
Fig. 4 illustrates such a score decomposition.

The scores are actually reported as the difference between the
student's scores and the maximum attainable for a test with a specified
length. The overall gain is that maximum less the original score. The
gain attainable from more information is that maximum less the revised

Digest 1   03/09/73 13.48.40 W.L. Sibley
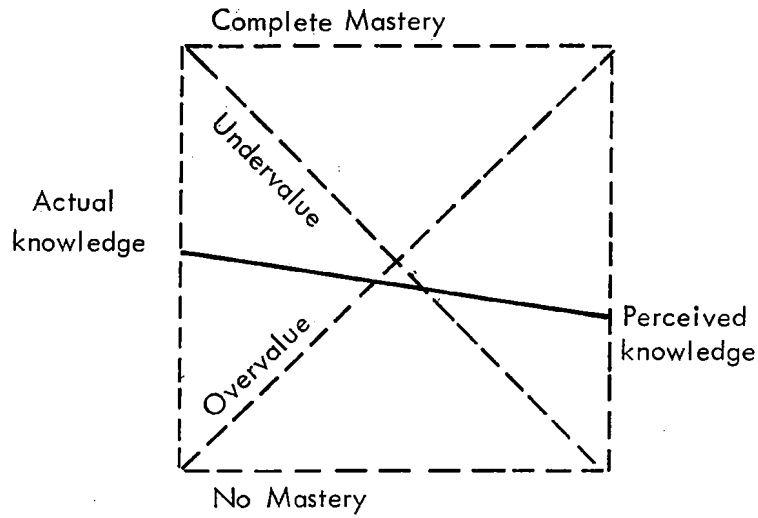
You tend to undervalue your knowledge.

Bet more heavily on answers you feel are correct.

You can improve your score by 37 points
by more realistic use of your knowledge.
You can improve your score by 224 points by more study.

You can improve your score by 261 points overall.

Fig. 4 — Score interpretation display

score. The gain from more "realistic" use of probabilities is simply the revised score less the original score.

INFORMATION THEORETIC MEASURES

The upper portion of Fig. 4 brings us to another interesting aspect of the logarithmic scoring system. We call the square in the upper part of Fig. 4 the student's information square. The term "information" arises from a very interesting relation between the logarithmic scoring system and the concept of "entropy" in information theory. That relation is developed below.

Equation (6) below is the expected score for a given question in our three alternative systems.

$$\text{Expected score} = \sum_{i=1}^{3} p_i (50 \log 3 \; p_i)$$

$$= 50 \left[\log 3 - \left(- \sum_{i=1}^{3} p_i \log p_i \right) \right] \tag{6}$$

Equation (7) is called, in information theory, the "entropy" of the partition p_i. It represents the expected amount of information which will be conveyed by the event itself; or, in other words, the expected surprise in finding which alternative is correct.

$$\text{Entropy} = - \sum_{i=1}^{3} p_i \log p_i \tag{7}$$

Equation (8) is the expected score normalized to the range 0 to 1.

$$\text{Relative information} = \frac{50 \left[\log 3 - \left(- \sum_{i=1}^{3} p_i \log p_i \right) \right]}{50 \log 3} \tag{8}$$

Equation (8) can be interpreted as follows: If there is no surprise content in the event (e.g., $p_1 = 1$, $p_2 = 0$, $p_3 = 0$) then there is perfect information (Eq. (8) = 1); if there is maximum surprise content in the event (i.e., $p_1 = p_2 = p_3 = 1/3$) then there is no information (Eq. (8) = 0). The upper part of Fig. 4 plots on the PERCEIVED axis the points corresponding to the relative information computed from the student's original probability distributions. The relative information derived from the probabilities as transformed by the realism line (the regression line in the external validity graph) is plotted on the REVISED axis and the two points are connected by a line. This information square has an interesting interpretation in terms of a proverb as depicted in Fig. 5.
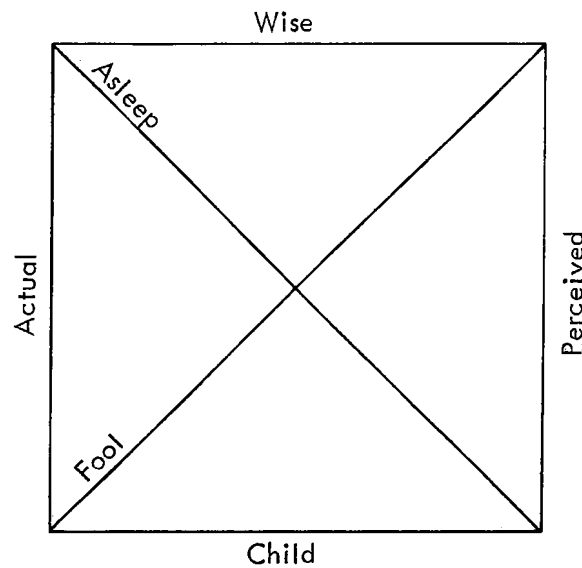
He who knows, and knows that he knows,
    He is wise, follow him.
He who knows, and knows not that he knows,
    He is asleep, awaken him.
He who knows not, and knows not that he knows not,
    He is a fool, shun him.
He who knows not, and knows that he knows not,
    He is a child, teach him.

Fig. 5 — A proverb

II. COMPUTER SYSTEM DESCRIPTION

HARDWARE

The current version of the computer-administered probability measurement program runs on The Rand Corporation's IBM 360 Model 65 computer, which is coupled to user consoles (see Fig. 6) via the Rand Video Graphic System.[4] Each videographic console consists of a studio-monitor-quality, black-and-white television set, a keyboard, and a hand-held light pen. The videographic light pen is unique in the sense that it counts television scan lines and light positions within those scan lines to report x,y positions within the display. This x,y position is transmitted from the console to the IBM 360/65 via the videographic system and is there compared to known positions or figures on the display. Pictures are constructed within the IBM 360/65 and transmitted to the consoles again via the videographic system. Figures 7a and 7b provide a system diagram for Rand's Video Graphic System.

SUPPORT SOFTWARE

The IBM 360/65 operates under IBM's operating system with the MVT (multi-tasking with a variable number of tasks) option. The operating system software has been extended by Rand to include the video operating system[5] which accommodates the message traffic between the 360/65 and the videographic terminals.

The probability measurement program is written in FORTRAN IV. The communication between the program and the videographic system is handled by a FORTRAN-callable graphics package called VPACK,[6] which was created especially for the Rand Video Graphic System. (Appendix A extracts the features of VPACK used by the measurement program.)

The present version of the measurement program does not allow for multiple consoles. It is submitted as a standard computer job which occupies an operating system partition for as long as testing continues. A multiple-console version would require additional hardware (more light pens) and extensive software development (a timesharing subsystem in IBM's OS/360).

Fig. 6 — Video Graphic console
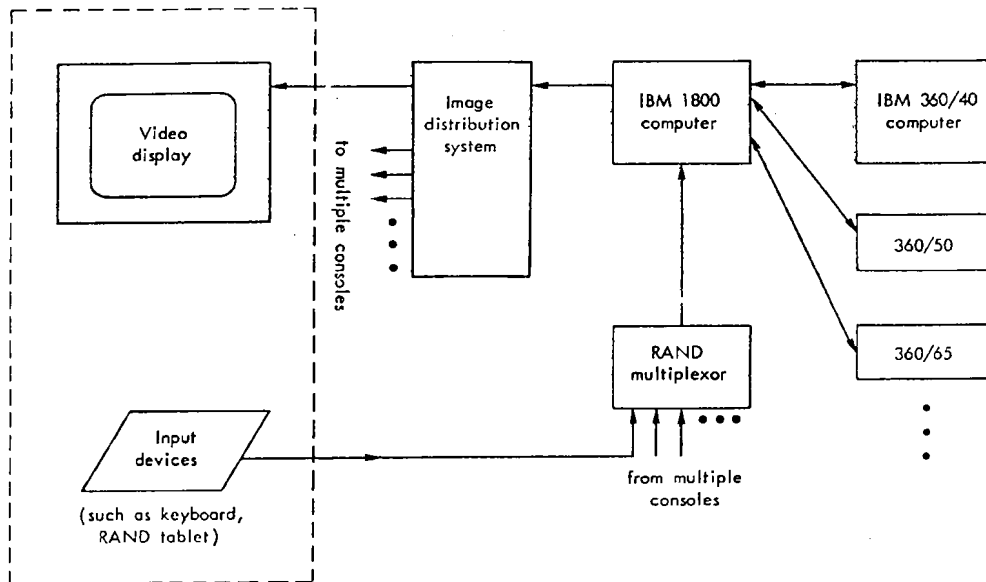
Fig. 7a — Video Graphic hardware configuration
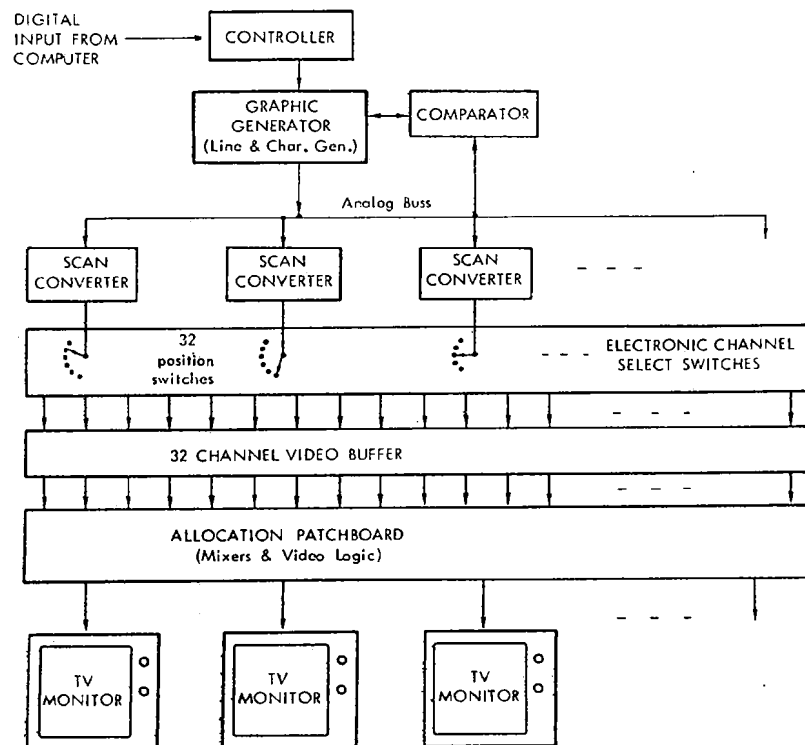
SOURCE: Ref. 3.

Fig. 7b — Image distribution system

SPECIFIC SOFTWARE

     The basic organization of the measurement program is illustrated in the state diagram in Fig. 8. The names within parentheses (for example, "SELECT") are the names of the subroutines called by the main program. (A complete listing of the main program and each of its subroutines is contained in Appendix B.) Each of the main subroutines manages a particular display associated with it. We will discuss the operation of the testing program in terms of each of these main subprograms. Each subroutine has the same logical structure; that is, each constructs a display, then waits for an action by the user with the light pen. The routine then compares the location of the light pen with various figures drawn in the display and either modifies the display to reflect the most recent action or returns control to the
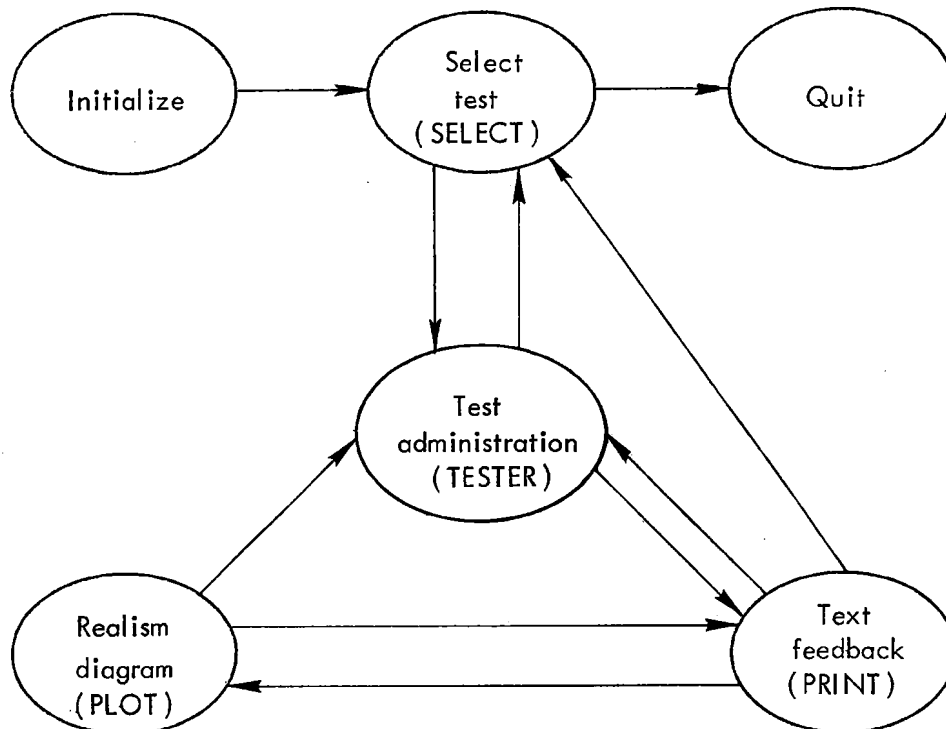
Fig. 8 — State diagram

main program, which then transfers to another state. In general, the user changes from one state to another by selecting a boxed area on the face of the display with his light pen. These areas are referred to as light buttons.

The main program serves the following functions:

- It collects the list of test names that are contained in the system files.
- It connects the program with the terminal.
- It asks the subject to submit his name.
- It switches among the main subroutines.
- It disconnects the program from the terminal when testing is complete.

After the subject has typed his name to identify himself to the system (responding to the display "PLEASE TYPE YOUR NAME:"), control is passed immediately to the test selection subroutine.

The test selection subroutine presents the user with the display illustrated in Fig. 9. The two boxes in the lower right-hand corner of Fig. 9 illustrate the light buttons mentioned above. If the user points to the word "QUIT," the test selection subroutine will return to the main routine with the indication that the main routine is to vacate the operating system partition and the computer. If the user selects the "TEST" light button, the program will transfer control to the test administration subroutine if a test has already been selected. If no test has been selected, the system will simply remain in the same state. By pointing at a box immediately to the left of the name of a test (illustrated on the left half of Fig. 9), the user indicates that he would like to take that particular test. The system then recovers that test information from the computer file and transfers control to the test administration subroutine. When a new test is selected, all of the information normally collected by the test administration and data-processing routines is reinitialized so that no information is kept between test administrations.

AVAILABLE TESTS

DIGEST 1
DIGEST 2
DIGEST 3
DIGEST 4
DIGEST 5
DIGEST 6
DIGEST 7
DIGEST 8
DIGEST 9
DIGEST 10
HUMANITIES A
HUMANITIES B
HUMANITIES C
HUMANITIES D
HUMANITIES E
HUMANITIES F
HUMANITIES G
NATURAL SCIENCES A
NATURAL SCIENCES B
NATURAL SCIENCES C
NATURAL SCIENCES D
SOCIAL SCIENCES A
SOCIAL SCIENCES B
SOCIAL SCIENCES C
SOCIAL SCIENCES D
SOCIAL SCIENCES E
SOCIAL SCIENCES F
SOCIAL SCIENCES G

PLEASE SELECT A TEST.

QUIT

TEST

Fig. 9 — Test selection display

Figure 10 illustrates one of the two displays presented by the
test administration subroutine. The test used in the figures is a
vocabulary test. The word to be defined in question 1 is "AUSTERITY."
The three allowed synonyms are "A: EXTREME SUFFERING," "B: POVERTY,"
and "C: SEVERE SIMPLICITY." For any given question, the console user
may select a point within the triangle labeled A, B, C to indicate the
probabilities he associates with each of the answers labeled A, B, C,
in the upper right-hand corner of Fig. 10. The triangle A, B, C is an
equilateral triangle and for any interior point, the sum of the dis-
tances from that point to each of the sides is a constant and can be
scaled to 1. In this way, the position of any point in the interior
of the triangle can be considered to be a selection of three probabil-
ities, each the probability of correctness associated with each of the
answers A, B, C. When the user selects a point within the triangle,
a small "x" appears at that point (see Fig. 10), and the logarithmic
scoring system is evaluated, using the three probabilities. The upper
left-hand corner display in Fig. 10 indicates what scores would apply
to each of the answers A, B, C, for the particular response represented
by the x. The score is presented in two ways: one, by the numeric rep-
resentation of the score; the other by the pictorial representation. The
user is free to reselect points within the triangle until he is satis-
fied that the scores displayed in the upper left-hand corner represent
his best bet for each of the possible responses.

Figure 11 illustrates the choice of option C with full certainty.
Figure 12 illustrates the situation in which a student has no reason
to choose one alternative over any other. The scores would vary
analogously for apex A or for apex B. When he is satisfied with his
selection, he may do one of several things as represented by the light
buttons in the lower right-hand corner of Fig. 10: (1) He may ask for
the next question; (2) he may ask for the previous question; (3) he may
ask for the answer to his question; or (4) he may return to the test
selection procedure.

Whenever he selects option (3), he will be presented with Fig. 13,
which illustrates the results up to the seventh question. The box in
the upper left-hand corner around the score for response A indicates

1. AUSTERITY

A   B   C

A: EXTREME SUFFERING

B: POVERTY

C: SEVERE SIMPLICITY

-17   -20   16
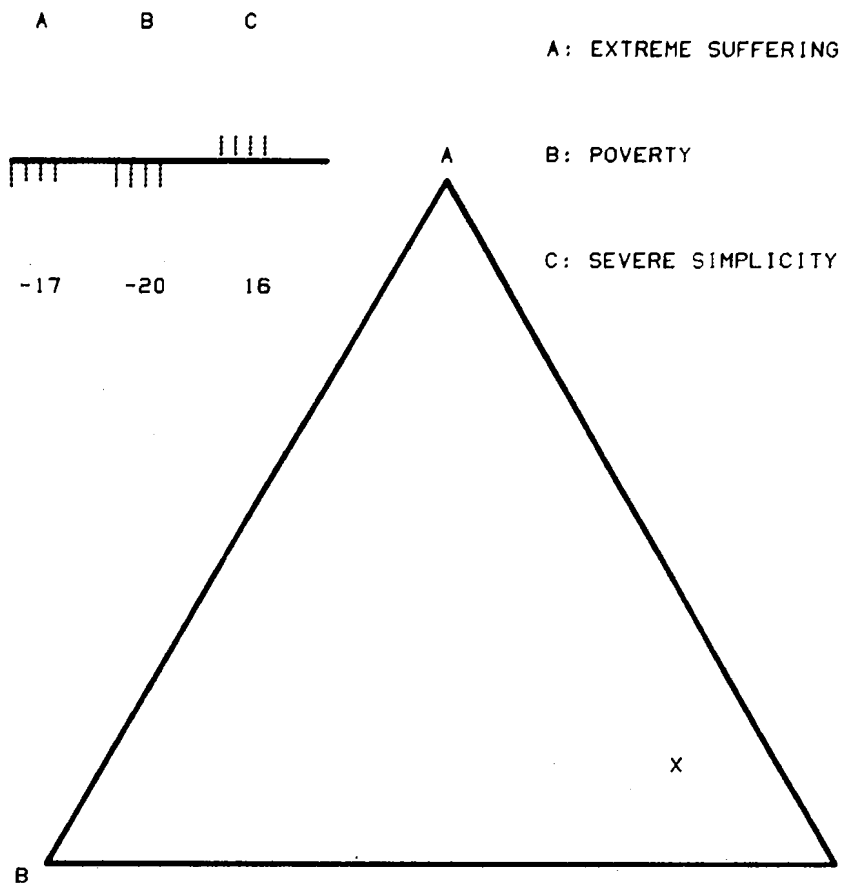
Fig. 10 — A selected point and associated scores

I. AUSTERITY

A   B   C

A: EXTREME SUFFERING

B: POVERTY

C: SEVERE SIMPLICITY

-76   -76   24

A

B   C
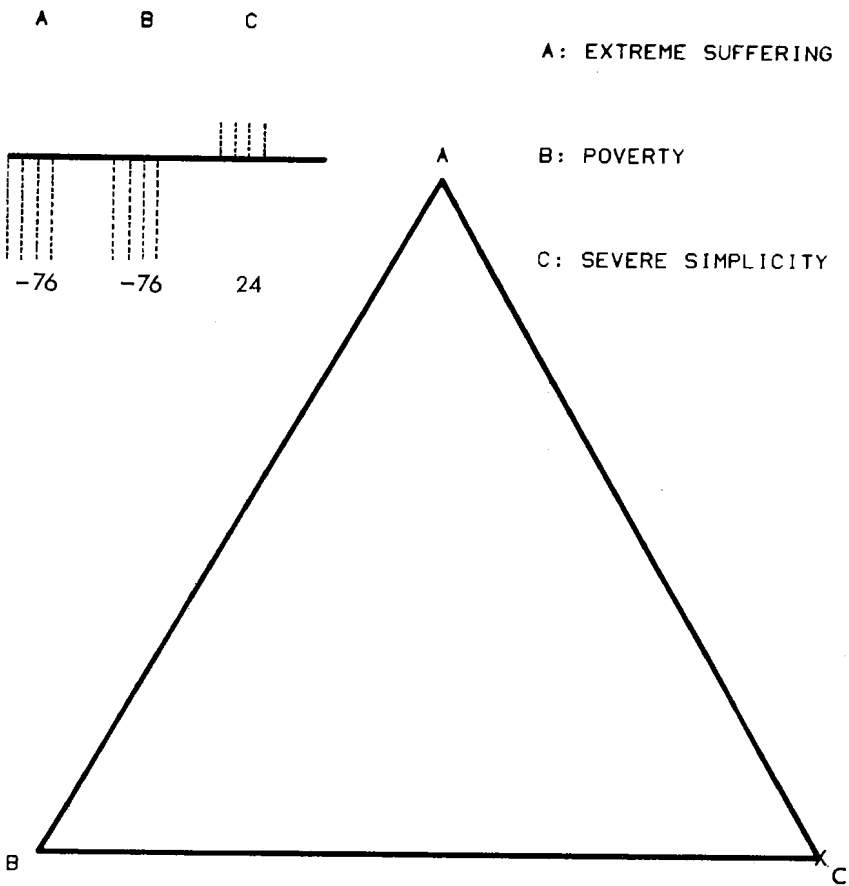
TEST
SELECT

QUESTION
ANSWER

PREVIOUS
QUESTION

NEXT
QUESTION

Fig. 11 — Choice of alternative "C"

I. AUSTERITY

A   B   C

A: EXTREME SUFFERING

B: POVERTY

C: SEVERE SIMPLICITY

0   0   0

```
                          A

                        /  \
                       /    \
                      /      \
                     /        \
                    /          \
                   /    x       \
                  /              \
                 /                \
                /                  \
            B  /_____\  C
```
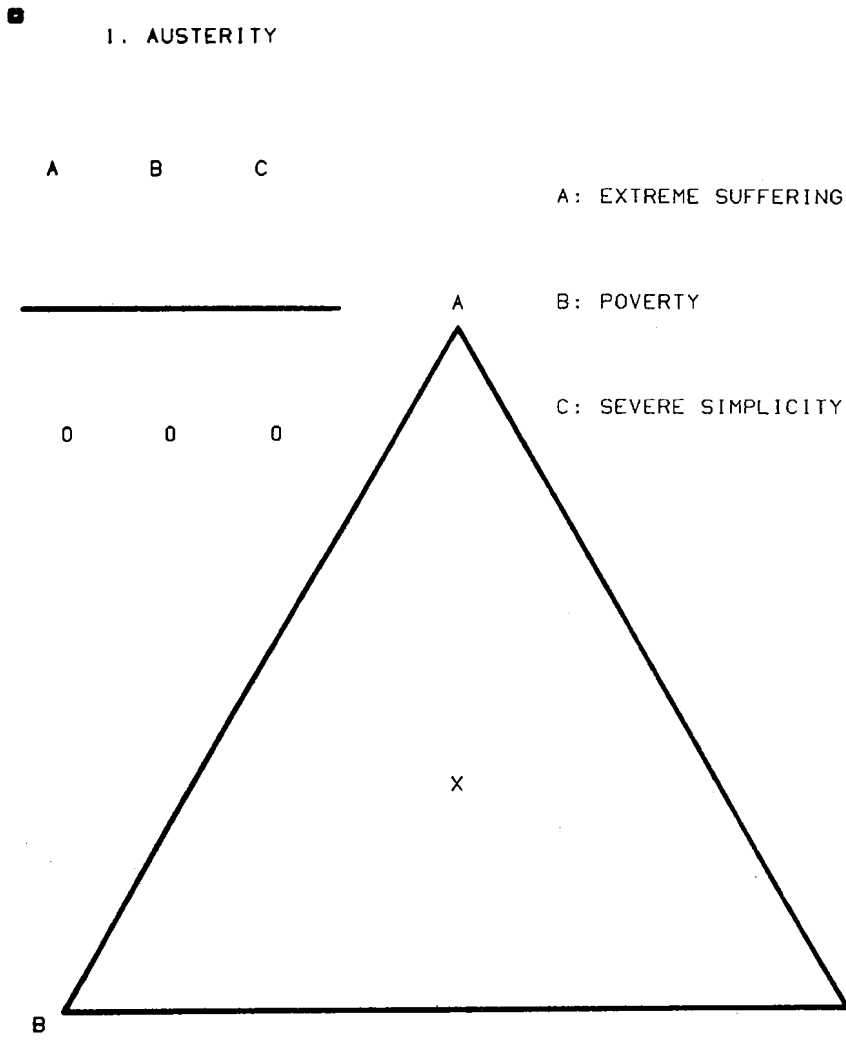
TEST
SELECT

QUESTION
ANSWER

PREVIOUS
QUESTION

NEXT
QUESTION

Fig. 12 — No preference for any alternative

7. COVEY

A: FLOCK

B: BLANKET

C: ESCORT

19   -28   -24

MAXIMUM POINTS

YOUR POINTS

MINIMUM POINTS
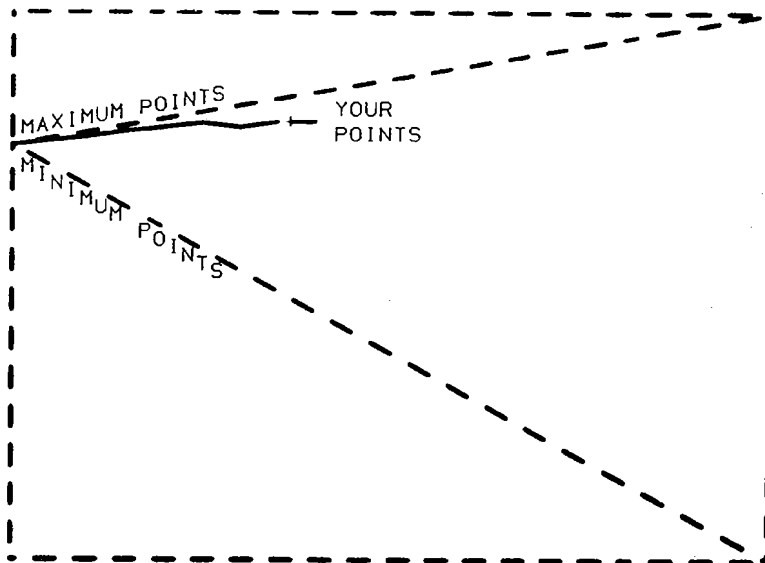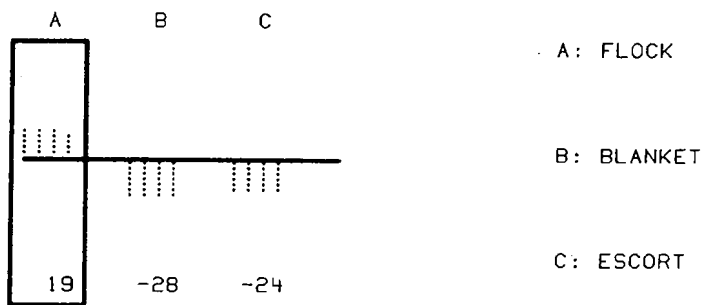
TEST SELECT

QUESTION ANSWER

PREVIOUS QUESTION

NEXT QUESTION

Fig. 13 — Cumulative score after seventh question

which of the three answers was correct and the numbers below indicate
how many points were gained or lost as a result of the user's assign-
ment of probabilities. The triangle has been replaced by a rectangle
which contains a plot of the cumulative score the user has attained as
he has gone from question to question. The line sloped to the upper
right is the line he would follow if he had assigned a probability of
1 to each of the correct answers. The line sloping down to the right
is the line he would follow if he had assigned the minimum probability
0.01 to the correct answer for each question. Once the user has asked
to see the answer to a question, he may no longer change the proba-
bilities he has associated with the answers. That is, if he returns
to the probability assignment display, he may no longer select a new
point in the triangle. If he attempts to, the system will respond with
a message that indicates that he already knows the answer to that
question.

If the user selects option (1) (the NEXT QUESTION), he is pre-
sented with that question and its associated responses. If the user
selects option (4) (TEST SELECT), he returns to the test selection pro-
gram with Fig. 13 displayed. He may select a new test, thereby de-
stroying the information he has collected to this point. Or he may
simply return to the present test with no information lost. If the
user has left some questions unanswered by the time he reaches the end
of the test, the system returns to the unanswered questions in turn
until each question has been answered. When the final question has
been answered, "TEST SELECT" changes its name to "SCORE PAGE" (see
Fig. 1). Figure 2 illustrates the answer display when all questions
have been answered. In this particular case, the vertical line with
"END" above it coincides with the right boundary of the plot. If there
were fewer than twenty questions in the test, the vertical line would
appear in the interior of the plot. If the user now selects "SCORE
PAGE," the system then transfers control to the score interpretation
page.

The score interpretation subroutine computes the variance of the
distribution of the probabilities and the realism parameters, that is,
the slope and intercept of the realism line. It then computes the

perceived and actual information measures and constructs the upper half
of the display illustrated in Fig. 4. Actual knowledge is plotted from
0 to 1 on the left vertical edge and perceived knowledge is plotted from
0 to 1 on the right vertical edge of the square. The line joining these
two points helps the user visualize the relationship between his actual
knowledge and what he has perceived his knowledge to be. If the vari-
ance of the probabilities is less than 0.01, the system assumes that it
has too little information to estimate reliably the realism line and
indicates to the user that the test was too difficult for him. If this
is not the case, the system then computes a bias measure which is simply
the difference between the perceived information measure and the actual
information measure. It then constructs a verbal response which re-
flects this bias. Figure 14 indicates decision points for the selec-
tion of the various messages. The system then computes the original
score and the improved score, restates them in light of the number of
points to be gained rather than the actual number of points attained,
presents them as part of messages indicating what could be gained (1) by
improvement in realism, (2) by gaining more information, and then pre-
sents (3) the total combined.

After observing the display, the user may (1) return to the testing
procedure, (2) return to the test selection procedure, (3) request a
printed summary of his test results (see Fig. 15), or (4) request the
realism plot.

Both the printed summary and the realism plot are primarily of use
to the investigator rather than to the student. The summary contains
additional information. We estimate the percentage of correct answers
a student would have obtained under the classical testing method. For
each question, if the maximum assigned probability corresponds to the
correct answer, we count that answer as correct. If the student assigns
a maximum probability to several responses, including the correct one,
we cannot treat the situation directly, so we treat it probabilistically
and assign a value of $\frac{1}{E}$ to this question, where E is the number of times
the maximum probability is used in that question. The average value of
these assignments over all questions represents an estimate of what the
student would have obtained had he been restricted to discrete choices.
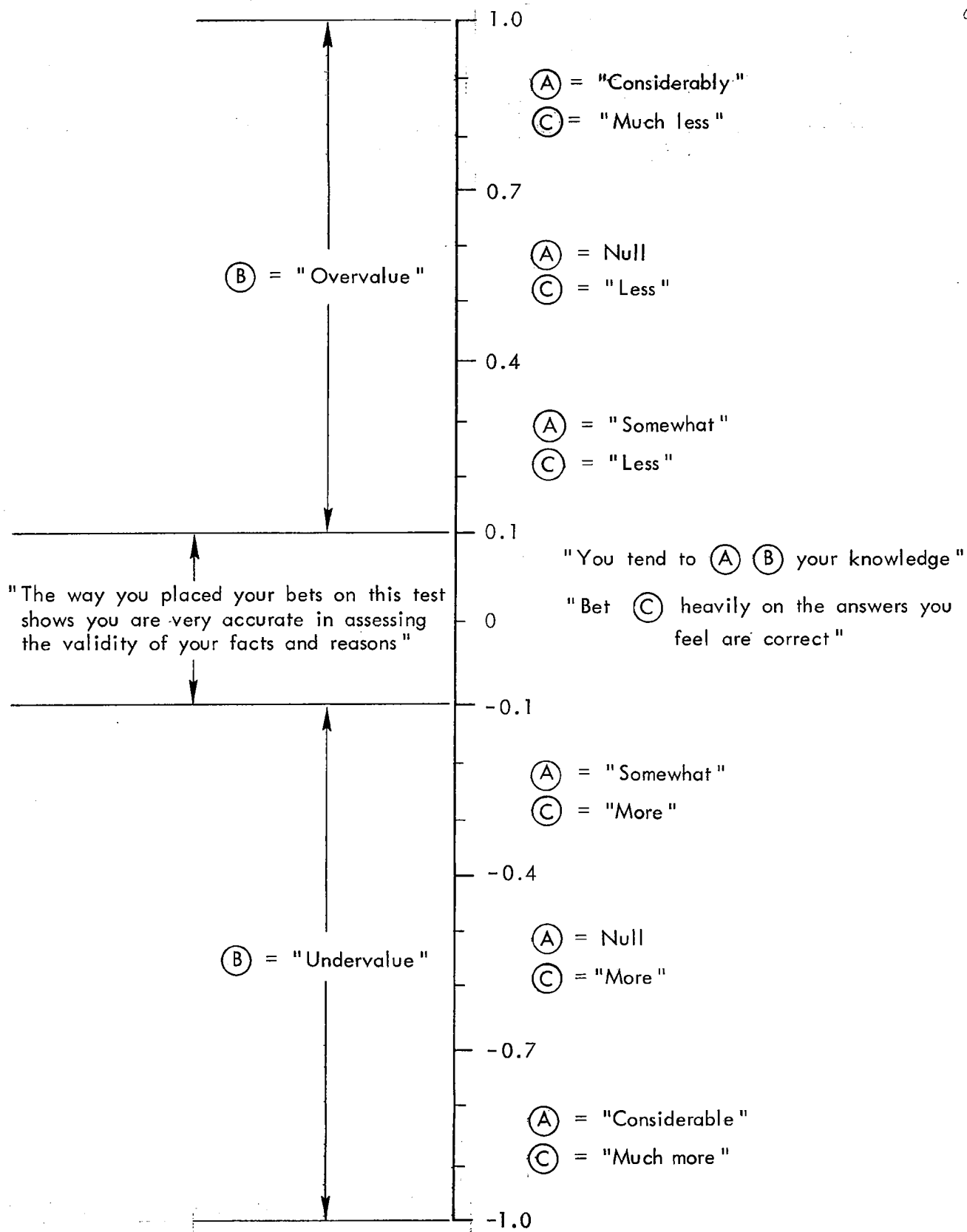
Fig. 14 — Bias decision points

```
TEST:DIGEST 1                    SUBJECT:W.L.SIBLEY

DATE:03/09/73      TIME:13.49.39

SCORE:  219     REVISED SCORE:  256    STEREOTYPE SCORE:  308

INFORMATION: 0.37          REVISED INFORMATION:  0.51

REALISM SLOPE:   1.23      REALISM INTERCEPT:  -0.08

BASE SLOPE:   0.85         BASE INTERCEPT:   0.05

BIAS:-0.15                 PROB. VARIANCE:   0.09
```

```
        PROBABILITIES                    SCORES
   A       B       C            A         B         C
  0.16    0.15    0.69**     -16.20    -17.27    16.05**
  0.13    0.72**  0.15       -19.65     16.82**  -17.53
  0.77**  0.11    0.12        18.23**  -23.42    -21.93
  0.15    0.14    0.71**     -17.07    -18.61     16.52**
  0.15    0.68**  0.17       -17.07     15.69**  -14.83
  0.73    0.12    0.15**      17.07    -21.68    -17.01**
  0.80**  0.09    0.11        19.25**  -28.90    -24.16
  0.70**  0.13    0.17        16.22**  -19.78    -14.75
  0.16    0.15    0.69**     -15.99    -16.81     15.90**
  0.71**  0.14    0.15        16.56**  -18.52    -17.32
  0.16    0.67**  0.17       -15.57     15.17**  -14.16
  0.0     0.0 **  1.00       -76.00    -76.00**   24.00
  0.15    0.70**  0.16       -17.74     16.14**  -16.13
  0.75**  0.12    0.13        17.79**  -22.71    -19.92
  0.09    0.11    0.80**     -27.36    -24.39     19.11**
  1.00**  0.0     0.0         24.00**  -76.00    -76.00
  0.13    0.13    0.74**     -19.90    -19.90     17.32**
  0.13    0.71**  0.15       -19.90     16.68**  -16.65
  0.77**  0.11    0.12        18.36**  -23.87    -22.35
  0.14    0.72**  0.15       -18.92     16.75**  -17.87
```

Fig. 15 — Printed test summary

This proportion of correct answers can be considered a base value z and a base line drawn through the points (1/3, 1/3) and (1,z). We also develop a STEREOTYPE score by using the probability z if the correct answer were assigned the maximum probability, (1-z)/2 otherwise. Finally, we compute a BIAS indicator which is simply the perceived information less the revised information. The asterisks indicate the correct answer in the columns under "PROBABILITIES" and "SCORES."

The realism plot subroutine constructs a display of the external validity graph with the associated plotted lines (Fig. 16). The vertical lines in the display are a histogram of the frequency of use of each of the probabilities along the bottom of the diagram. The class interval for this histogram is 0.1. The x's distributed throughout the display are the relative frequencies with which elements in those class intervals were associated with the correct answer. The solid line going from the lower left to the upper right-hand corner of the display is the ideal realism line. The solid line at an angle to the ideal line is the fitted realism line and the dashed line is the base line. From this point the user may return to the score page or the test procedure.

Since this version of the measurement program is primarily an exploratory one, the software was designed to make that exploration as easy as possible. The various functions of test selection, test administration, and the two feedback processes are all isolated to allow changes to be made in each without affecting the others. The logic in each one of the major subroutines is as simple as possible to allow ease of change.
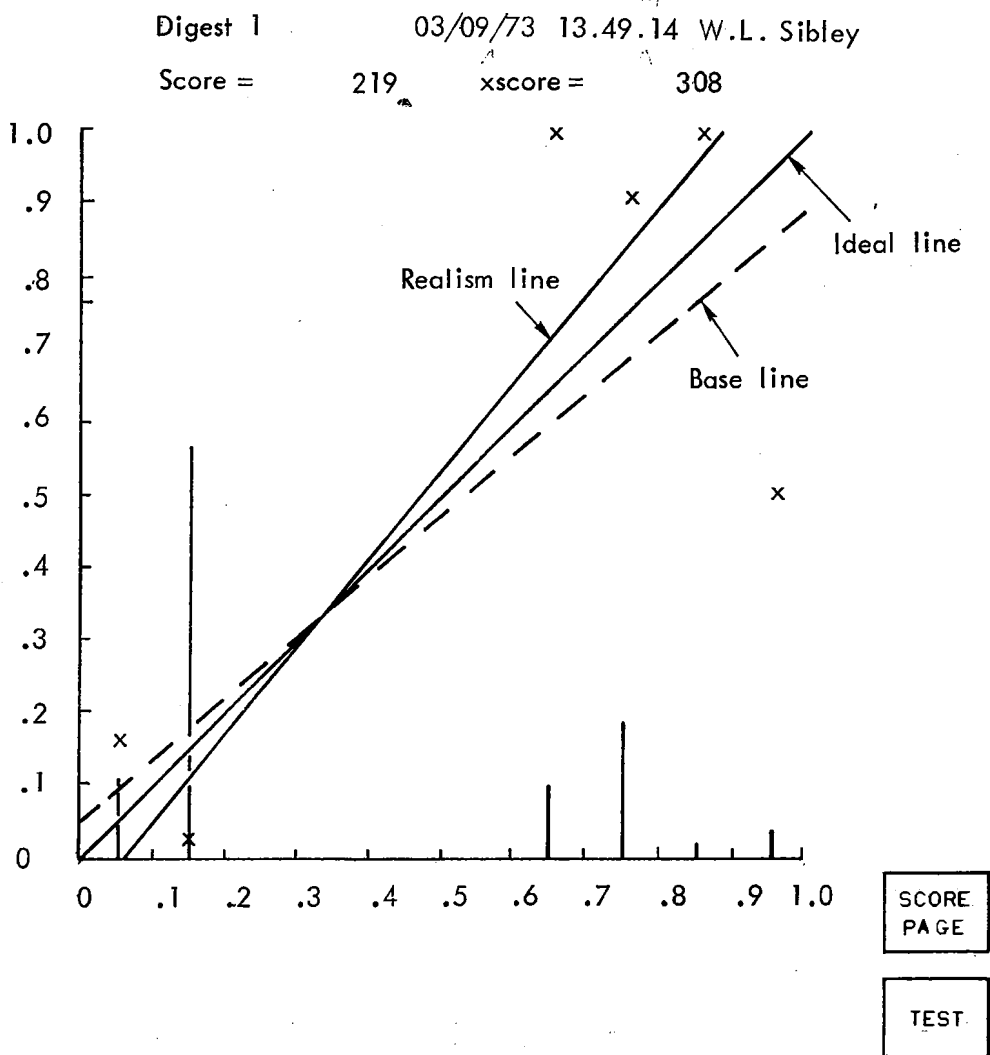
Fig. 16 — External validity graph (realism plot)

III. PLANNED MODIFICATIONS AND EXTENSIONS

PHILOSOPHY OF USE

The general philosophy of the present form of the testing program is that everything should be as simple and as easy to use as possible. The user should not be concerned about the fact that he is using the computer and that fact should not overawe him. Aside from considerations of the responsiveness and stability of the system, a major concern is with the ability of the student to grasp the relation between his score and his actions with the triangle. He must be allowed a maximum amount of flexibility and the system should not encourage him to fall into the use of stereotyped responses. In addition, the system should be as nearly self-teaching as possible. That is, the actions to be taken should be as self-evident as they can be made. The extent to which the latter situation can be attained is now under investigation by observing, questioning, and recording the responses of users of the test procedure.

POSSIBLE MODIFICATIONS

Since textual messages are difficult to construct and are sometimes misleading (e.g., the bias and bet messages), it seems best to present the information to the student in some nonverbal form. One possible approach to such a presentation would be to eliminate a good part of the display on the score page and to allow the student to re-cycle through the test, having the system present in the triangle both his original response and the response computed by the system on the basis of the realism line (Fig. 16). This graphical display would illustrate to the student how he should modify his behavior to maximize his score with his present level of knowledge. The appropriate kind and amount of feedback and advice to be provided the student will be the major focus of our research in the immediate future.

The present system has no provision for permanently collecting data in machine readable form on the student's responses. As we gain experience with the data represented by Fig. 17, we will design and

20. PERSPICACITY

A   B   C

A: SUSPICION

B: ACUTE MENTAL VISION

C: PERSISTENCE

-18   16   -17

SCORE
PAGE

QUESTION
ANSWER

PREVIOUS
QUESTION

NEXT
QUESTION
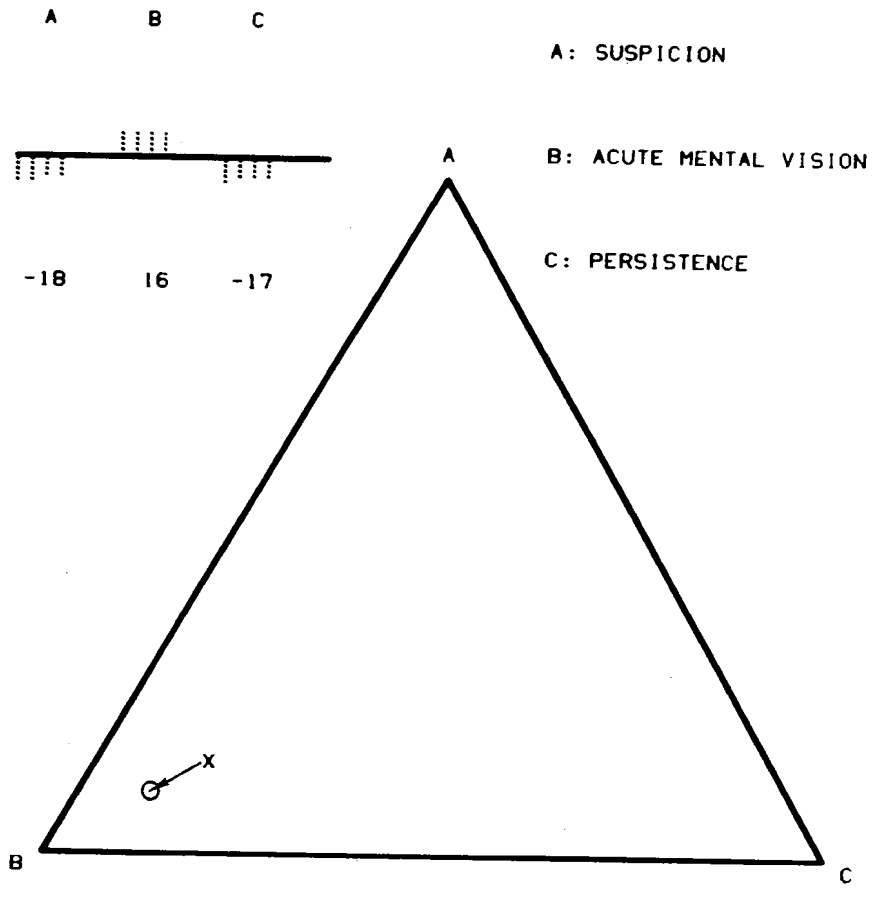
Fig. 17 — Adjusted response

implement a scheme for managing the collection of data over many students and many tests and for performing appropriate statistical analyses of the results. In fact, an important goal of our work is to provide the ultimate user of admissible probability measurement with a complete set of tools for data collection, analysis, and interpretation.

IV. PLATO IV

The PLATO IV system is designed to be an economically viable large-scale, computer-based education system.[7]   The computer-based Education Research Laboratory at the University of Illinois has developed versions of the system from a single terminal connected to a 1954 vintage computer to a collection of 20 graphic-pictorial terminals connected to a more powerful computer. Over the period of development the laboratory has gathered data on more than 70,000 student contact hours. The present version (PLATO IV) is designed to bring the terminal/computer costs to within the 25¢ to 30¢ range per student contact hour.

A PLATO IV terminal is an interactive computer graphics terminal with the following features:[8]

1. An $8\frac{1}{2}$ inch square plasma display panel that is readable in a brightly lighted room without eyestrain.
2. Permanent storage of information on the display screen without flicker. Absolutely no refreshing of the display panel by the computer is required. Selective WRITE and ERASE capability exists for all dots, lines, and characters.
3. Self-contained character and line generators.
4. A character writing speed of 180 characters per second and the capability of displaying 2048 characters on the screen.
5. A line drawing speed in excess of 600 inches per second.
6. A character repertoire of 252 characters, 126 of which are alterable through the computer program.
7. The ability to transmit and receive data on voice trade telephone circuits.
8. A random-access slide projector for rear projection of static information on the display screen.
9. Additional input-output channels for the control of auxiliary equipment.
10. An optional random-access audio response unit.

11. A pointing matrix for the surface of the display. An inter-
     ruption of light beams from two edges of the display to the
     opposite edges determines a point in the matrix. That point
     is reported to the computer and correlated with the contents
     of the display. The matrix has 16 divisions in each dimension
     of an $8\frac{1}{2} \times 8\frac{1}{2}$ inch area.

PLATO IV is designed to support remote classrooms consisting of
up to 32 terminals. The data rate to each terminal varies from $2\frac{1}{2}$ to
80 10-bit characters a second, depending on the activity of the 32
terminals.

The PLATO IV hardware and software are designed specifically to
deal simultaneously with large numbers of students, each following his
own course of study. A course author (in our case, a test author) writes
his course material in the TUTOR language.[9] TUTOR allows the author
to structure his material easily to accommodate a wide variety of stu-
dent responses. In addition, the language includes facilities for data
collection and computation as well as access to files of course materials.

The capabilities of the PLATO IV system and the availability of
terminals in the military services provide a vehicle for extensive test-
ing of the admissible probability measurement concept in the military
environment. Our intent is to develop a PLATO IV version of admissible
probability measurement, including appropriate data collection, analysis,
and interpretation as a package for the military services. We hope to
encourage and assist the services to incorporate computer-based admis-
sible probability measurement in their education and training programs.

Appendix A
SUPPORT SOFTWARE

VPACK ROUTINES

The VPACK display routines operate on a buffer provided by the calling program. The programmer specifies the address and size of the buffer during initialization. Thereafter, VPACK retains this information so that the programmer no longer need be concerned with it. VPACK also maintains information about the amount of buffer space currently being used (the *current length*) and the position into which the display commands are being stored (the *current position*). These data are automatically updated by the several routines as appropriate. (See Fig. 18.) The current length of a buffer may exceed the maximum length allowed by the Video Graphic System (VGS) for a single picture segment. In this event, VPACK automatically segments the picture at appropriate points and passes the multiple segments to VGS.

More detailed descriptions of the subroutines follow, in alphabetical order.

VCHAR: Adds a character string to the buffer.

CALL VCHAR (IX, IY, ST, N)

VCHAR adds N characters from the string ST to the buffer, beginning at the current position. IX and IY specify the center of the first character position. The intensity of the string is set to normal. The current position and current length are updated.

VCOPY: Produces hardcopy of the current picture.

CALL VCOPY (L)

VCOPY effectively produces hardcopy (both paper and 35mm film) of the current picture. More specifically, it reformats the buffer into a meta language, which it writes on a disc data set named SC4060ZZ.
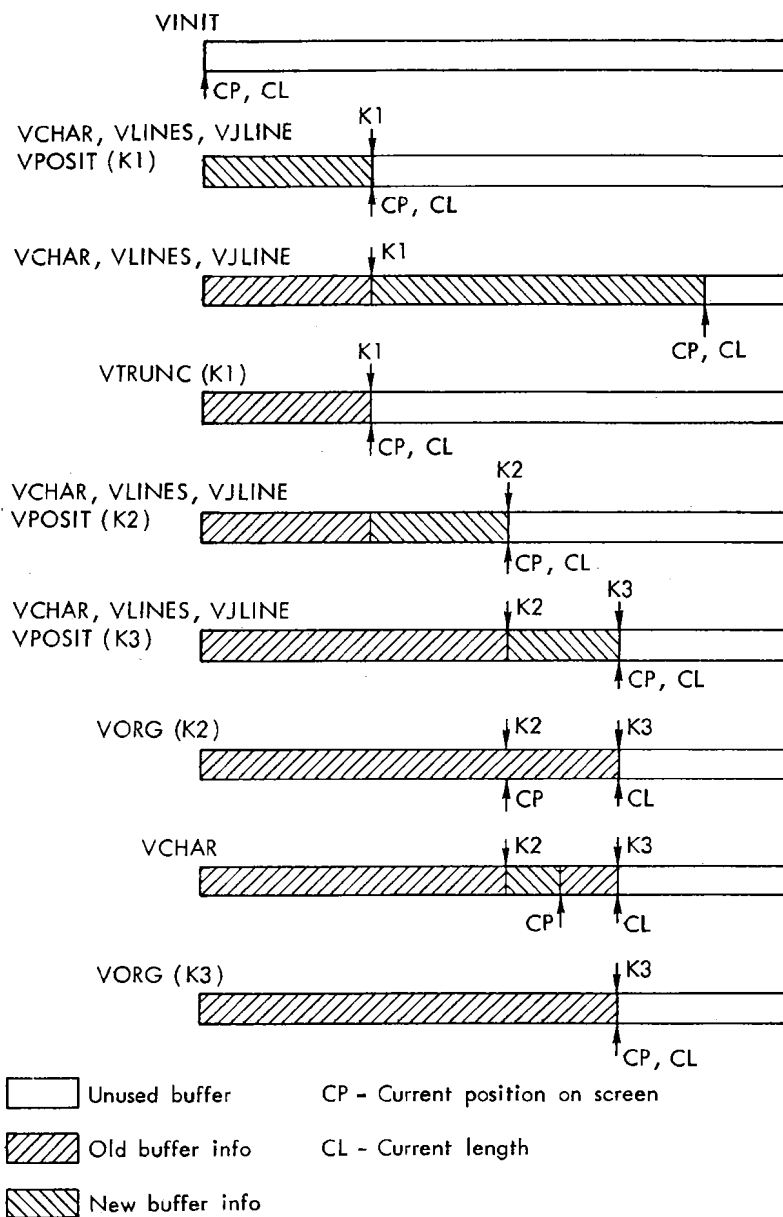
Fig. 18 — Use of current position and current length

This data set is subsequently transferred to magnetic tape and processed off-line.

Parameter L specifies the action to be taken by the microfilm processor after drawing the image. Its values are:

0   Produce 11 × 14 copy and advance the frame

1   Produce 11 × 14 copy but do not advance the frame

2   Produce 8 × 11 copy and advance the frame

3   Produce 8 × 11 copy but do not advance the frame

4   Close the output file

In normal operation, the frame is advanced after each frame is drawn. If the frame is not advanced, the next picture that is copied will be superimposed onto the current picture.

VINIT: Initializes the display, tablet, and keyboard.

CALL VINIT (NAME,BUFF,N)

VINIT connects the program to the VGS and activates the tablet and keyboard. NAME is a string of exactly eight characters which the user will type to connect his terminal to the program (CREATE ↔ NAME). If the name consists of fewer than eight characters, it must be padded with trailing blanks. BUFF is the buffer, which VINIT initializes with a RESET command. N is the maximum length of BUFF in bytes. The current position and current length are initialized.

An appropriate value for N can be estimated by considering the longest buffer that will be created, using the following guide:

● Each call to VCHAR requires n + 6 bytes, where n is the length of the character string.

● Each call to VLINES requires 10 n + 1 bytes, where n is the number of discrete line segments.

● Each call to VJLINE requires 5 n + 6 bytes, where n is the number of joined line segments.

- Each call to VNULL requires n bytes, where n is the number of null codes to be added.
- Each call to VSIZE, VLTYPE, or VDIM requires 1 byte.
- All other calls require no space.

VJLINE: Adds joined line segments to the buffer.

CALL VJLINE (N,IX,IY)

VJLINE adds N joined line segments to the buffer, beginning at the current position. The lines are set to normal intensity. The current position and current length are updated.

The I*th* line extends from the point IX(1), IY(1) to the point IX(1+1), IY(1+1). Arrays IX and IY must therefore contain at least N + 1 entries. All coordinates after the first are entered in the buffer relative to the first; this enables the program to move the entire set of lines by a single call to VMOVE.

VLINES: Adds discrete line segments to the buffer.

CALL VLINES (N,IXF,IYF,IXT,IYT)

VLINES adds N discrete line segments to the buffer, beginning at the current position. The line segments are set to normal intensity. The current position and current length are updated.

The I*th* line extends from IXF(I), IYF(I) to IXT(I), IYT(I). All coordinates after the first are entered in the buffer relative to the first; this enables the program to move the entire set of lines by a single call to VMOVE.

VLTYPE: Adds a line-type code to the buffer.

CALL VLTYPE (L)

VLTYPE adds a line-type code to the buffer at the current position. The current position and current length are updated. The line-type code is a model relative to the buffer; that is, once the line-type has been specified, it applies to all lines appearing in the buffer

until the next line-type code appears. If no line-type code is specified, solid lines are used. The values of L are:

1. Solid lines
2. Dashed lines
3. End points only

VORG: Resets the current buffer position.

CALL VORG (J)

VORG sets the current position to the value of J. The current
length is unchanged. The value of J must not exceed the current length.
This routine allows the programmer to reconstruct part of the display
by replacing a sequence of buffer orders with another sequence of the
same length. To guarantee that identical lengths are used, it is good
practice to replace character strings only with other character strings,
sequences of joined lines with other sequences of joined lines, etc.,
and to ensure that the number of characters or lines is the same as
the number being replaced.

VLPEN: Communicates with the light pen.

CALL VLPEN (IACT,ISIZE,ITIME,IDATA,IEXIT)

VLPEN is an adaptation of a routine which communicates with the
Rand Tablet and is used for handprinted symbol recognition studies.
The parameters of interest for the light pen are IDATA and IEXIT.
IDATA is a two-entry integer array which contains the (x,y) location
of a light pen strike. IEXIT is an integer which indicates the reason
for the return from the routine. VLPEN normally waits for a light pen
interrupt before returning control to the calling routine.

VPOSIT: Retrieves the current buffer position.

CALL VPOSIT (J)

VPOSIT returns the current position in location J. The current position and current length are unchanged.

VSEND: Sends a picture to the screen.

CALL VSEND

VSEND sends an updated picture to the screen. Because all of the other picture manipulation routines merely alter the buffer, this routine must be called to make their effects visible to the user. The current position and current length are unchanged.

VTRUNC: Truncates the buffer.

CALL VTRUNC (J)

VTRUNC sets the current length to the value of J. If the current position exceeds the current length. This routine effectively deletes all display code appearing in the buffer beyond the specified point.

FORTRAN CONVERSION ROUTINES

FORMAT: Converts a number to a character string.

CALL FORMAT (IFMT,IW,ID,VALUE,CHARS)

FORMAT converts a number, either real or integer, to a character string in I, F, or E format. IFMT specifies the type of conversion to be performed: (1) to convert an integer to I format; (2) to convert a real number to F format; or (3) to convert a real number to E format. IW is the field width of the converted number. ID specifies the number of decimal places; ID = 0 if IFMT = 1. VALUE contains the number to convert; it may be either real or integer, depending on the value of IFMT. CHARS will contain the resulting character string; it must be an array large enough to hold at least IW characters.

<u>CONVRT</u>: Converts a character string to a number.

      CALL CONVRT (IFMT,IW,STRING,OUT)

     CONVRT converts IW characters of the character string STRING to
a number, either real or integer. IFMT specifies the conversion:
(1) to convert to an integer; (2) to convert to a real; or (3) to con-
vert to a real if a decimal point or the letter E appears in the string,
but to an integer otherwise. On exit, OUT contains the converted value,
either real or integer.

PROGRAM LISTING

MAIN PROGRAM

```
C **** ARRAYS ****
      DIMENSION M1(7)
      DIMENSION QUEST(16,4,40)
      DIMENSION CHOICE(8,3,3,40),ANSW(40)
      DIMENSION RESPON(3,40),LOCKED(40)
      DIMENSION TEST(5),SUBJCT(16)
      DIMENSION IDATA(2)
      DIMENSION BUFF(1200),MARK(20),BLANKS(16)
      DIMENSION FILES(5,50)
C **** TYPE DECLARATIONS ****
      INTEGER M1,QUEST,ANSW,CHOICE,TEST,SUBJCT,FILES
      INTEGER SAVE,SCORE,LOCKED
C **** DATA ****
      DATA M1/'PLEA','SE T','YPE ','YOUR',' NAM','E   ','    '/
      DATA BLANKS/16*'    '/
      DATA MDIM/40/
C ************** FORMAT STATEMENTS *********************************
  100 FORMAT(I4)
  200 FORMAT(16A4)
C ************ READ NAMES OF AVAILABLE TESTS *****************
      READ (9,100) NFILES
      DO 300 I=1,NFILES
  300 READ (9,200) (FILES(K,I),K=1,5)
      REWIND 9
C ************* INITIALIZE ************************************
      INDEX=0
C ************ CONNECT TO THE TERMINAL *********************
      CALL VLINIT('TEST    ',BUFF,4700)
      CALL VPOSIT( MARK(1))
      CALL VEDSET(0,0,2,0)
C **** GET THE SUBJCT'S NAME ****
      CALL VORG(MARK(1))
      CALL VCHAR(0,500,M1,28)
      CALL VPOSIT(ITERM)
      CALL VTRUNC(ITERM)
      CALL VSEND
      CALL VEDIT(0,450,64,BLANKS,SUBJCT,ITERM)
C ********* SELECT THE TEST ********************************
  400 CALL SELECT(MDIM,INDEX,MARK,NFILES,FILES,QUEST,CHOICE,ANSW,
     X           RESPON,LOCKED,BLANKS,TEST,N,NEXT)
      GO TO (400,500,600,700,800), NEXT
C ********* DO THE TESTING *****************************
  500 CALL TESTER(MDIM,MARK,QUEST,CHOICE,ANSW,RESPON,LOCKED,
     X           INDEX,N,BLANKS,NEXT)
      GO TO (400,500,600,700,800), NEXT
```

```
C ************** PRINT THE RESULTS *********************
  600 CALL PRINT(MDIM,INDEX,N,RESPON,ANSW,MARK,BLANKS,TEST,SUBJCT,
     X           GEOM,PVAR,SCORE,SREAL,TREAL,SBASE,TBASE,NEXT)
      GO TO (400,500,600,700,800), NEXT
C ********** PLOT THE RESULTS ***************************
  700 CALL PLOT(MDIM,MARK,SCORE,GEOM,TEST,SUBJCT,SREAL,TREAL,
     X           SBASE,TBASE,N,RESPON,ANSW,NEXT)
      GO TO (400,500,600,700,800), NEXT
C ************ QUIT QUIT QUIT **************************
  800 CALL VTRUNC(MARK(1))
      CALL  VSEND
      CALL EXIT
C ****************************************************
      END
```

TEST SELECTION SUBROUTINE

```
      SUBROUTINE SELECT(/MDIM/,/INDEX/,/MARK/,/NFILES/,/FILES/,
     X               /QUEST/,/CHOICE/,/ANSW/,/RESPON/,
     X               /LOCKED/,/BLANKS/,/TEST/,/N/,/NEXT/)
      DIMENSION MARK(1),FILES(5,MDIM),QUEST(16,4,MDIM)
      DIMENSION CHOICE(8,3,3,MDIM)
      DIMENSION ANSW(1),BLANKS(1),TEST(1),RESPON(3,MDIM),LOCKED(1)
C ************* DIMENSION STATEMENTS *************************
      DIMENSION M2(6)
      DIMENSION IDATA(2)
      DIMENSION BOXX(5,50),BOXY(5,50)
C ************* TYPE DECLARATIONS ***************************
      INTEGER MARK,FILES,QUEST,CHOICE,ANSW,LOCKED,BLANKS,TEST
      INTEGER M2
      INTEGER IDATA,BOXX,BOXY
C ************* DATA STATEMENTS *****************************
      DATA M2/'PLEA','SE S','ELEC','T A ','TEST','.   '/
      DATA  BOXX/875,975,975,875,875,
     X           875,975,975,875,875,
     X           0,10,10,0,0,235*0/
      DATA  BOXY/10,10,85,85,10,
     X           110,110,185,185,110,
     X           990,990,999,999,990,235*0/
      DATA NARRAY/50/
C *************** FORMAT STATEMENTS *************************
  100 FORMAT(I4)
  200 FORMAT(16A4)
C ************ START BY GETTING THE TEST NAME ****************
      CALL VORG(MARK(1))
      CALL VCHAR(75,950,'AVAILABLE TESTS',15)
      DO 300 KK=1,NFILES
      CALL VCHAR(75,950-(KK+1)*24,FILES(1,KK),20)
      BOXX(1,KK+3)=0
      BOXX(2,KK+3)=50
      BOXX(3,KK+3)=50
      BOXX(4,KK+3)=0
      BOXX(5,KK+3)=0
      BOXY(1,KK+3)=950-(KK+1)*24-10
      BOXY(2,KK+3)=BOXY(1,KK+3)
      BOXY(3,KK+3)=BOXY(1,KK+3)+20
      BOXY(4,KK+3)=BOXY(1,KK+3)+20
      BOXY(5,KK+3)=BOXY(1,KK+3)
      CALL VJLINE(4,BOXX(1,KK+3),BOXY(1,KK+3))
  300 CONTINUE
      CALL VJLINE(4,BOXX(1,1),BOXY(1,1))
      CALL VCHAR(BOXX(1,1)+24,BOXY(1,1)+36,'TEST',4)
```

```
      CALL VJLINE(4,BOXX(1,2),BOXY(1,2))
      CALL VCHAR(BOXX(1,2)+24,BOXY(1,2)+36,'QUIT',4)
      CALL VCHAR(75,950-(NFILES+3)*24,M2,24)
      CALL VJLINE(4,BOXX(1,3),BOXY(1,3))
      CALL VPOSIT(ITERM)
      CALL VTRUNC(ITERM)
      CALL VSEND
C ********** WAIT FOR PEN ACTION *****************************
  400 CALL PEN(IDATA,BOXX,BOXY,NARRAY,NBOX)
      IF(NBOX.EQ.0) GO TO 400
      IF(NBOX.GT.3) GO TO 600
      IF(NBOX.EQ.2) GO TO 500
      IF(NBOX.EQ.3) GO TO 1500
C *********** RETURN TO TESTING *****************************
      IF(INDEX.EQ.0) GO TO 400
      NEXT=2
      RETURN
C ************ QUIT ****************************************
  500 NEXT=5
      RETURN
C ***** RESET INTERNAL RECORDS *****
  600 INDEX=1
      DO 700 I=1,3
      DO 700 J=1,MDIM
  700 RESPON(I,J)=0.
      DO 800 I=1,MDIM
  800 LOCKED(I)=0
C ***** INPUT TEST QUESTIONS AND ANSWERS *****
      DO 900 I=1,5
  900 TEST(I)=FILES(I,NBOX-3)
      CALL VORG(MARK(1))
      CALL VCHAR(500,500,TEST,20)
      CALL VPOSIT(ITERM)
      CALL VTRUNC(ITERM)
      CALL VSEND
      KK=NBOX+6
      READ (KK,100) N
      DO 1400 I=1,N
      DO 1000  J=1,4
 1000 READ (KK,200) (QUEST(K,J,I),K=1,16)
      DO 1100 J=1,3
      DO 1100 L=1,3
 1100 READ (KK,200) (CHOICE(K,L,J,I),K=1,8)
      CALL VORG(MARK(1))
      KKK=MOD(I,2)
      IF (KKK.NE.0) GO TO 1200
      CALL VCHAR(500,500,TEST,20)
      GO TO 1300
 1200 CALL VCHAR(500,500,BLANKS,20)
 1300 CALL VPOSIT(ITERM)
```

```
      CALL VTRUNC(ITERM)
      CALL VSEND
1400  READ (KK,100) ANSW(I)
      REWIND KK
      NEXT=2
      RETURN
1500  CALL VCOPY(2)
      GO TO 400
      END
```

TEST ADMINISTRATION SUBROUTINE

```
      SUBROUTINE TESTER(/MDIM/,/MARK/,/QUEST/,/CHOICE/,/ANSW/,
     X          /RESPON/,/LOCKED/,/INDEX/,/N/,/BLANKS/,/NEXT/)
      DIMENSION MARK(1),QUEST(16,4,MDIM),CHOICE(8,3,3,MDIM),
     X          ANSW(1),RESPON(3,MDIM),LOCKED(1),BLANKS(1)
C ******************* DIMENSIONS *************************
      DIMENSION BOXX(5,5),BOXY(5,5)
      DIMENSION TRIX(4),TRIY(5)
      DIMENSION IX(51),IY(51)
      DIMENSION PTS(5),SCR(3),M3(5)
      DIMENSION IDATA(2)
      DIMENSION MMAX(14),MMIN(14),THEANS(9)
C ******************* TYPE DECLARATIONS *****************
      INTEGER MARK,QUEST,CHOICE,ANSW,LOCKED,BLANK,SCORE
      INTEGER BOXX,BOXY
      INTEGER TRIX,TRIY
      INTEGER R1X,R1Y,R2X,R2Y,R3X,R3Y,QPOSX,QPOSY
      INTEGER PTS,SCR,M3,REVIE,THEANS
      INTEGER IDATAY,NARRAY
C ******************* INITIAL DATA **********************
      DATA TRIX/425,50,800,425/
      DATA TRIY/700,50,50,700/
      DATA BOXX/875,995,995,875,875,
     X          875,995,995,875,875,
     X          875,995,995,875,875,
     X          875,995,995,875,875,
     X          0,10,10,0,0/
      DATA BOXY/10,10,85,85,10,
     X          110,110,185,185,110,
     X          210,210,285,285,210,
     X          310,310,385,385,310,
     X          990,990,999,999,990/
      DATA QPOSX,QPOSY,R1X,R2X,R3X,R1Y,R2Y,R3Y
     X     /95,975,525,525,525,825,725,625/
      DATA PTS/'   ',' , ',' . ',' .. ',' ... ',' .... '/
      DATA M3/'QUES','TION',' # P','LEAS','E   '/
      DATA NARRAY/5/,MAXPLT/20/
      DATA MMAX/'M  ','A  ','X  ','I  ','M  ','U  ',' ',
     X          'M  ',' ',' ','P  ','O  ','I  ','N  ',' ',
     X          'T  ','S  '/
      DATA MMIN/'M  ','I  ','N  ','I  ','M  ','U  ',' ',
     X          'M  ',' ',' ','P  ','O  ','I  ','N  ',' ',
     X          'T  ','S  '/
      DATA THEANS/'SORR','Y, Y','OU A','LREA','DY K','NOW ',
     X            'THE ','ANSW','ER. '/
C ********* THE SCORING FUNCTION ***********************
```

```
      VALUE(X)=50.*ALOG10(3.*AMAX1(.01,X))
C ****************** DRAW THE BOXES ****************************
      CALL VORG(MARK(1))
C **** DRAW THE 'NEXT' BOX ****
      CALL VJLINE(4,BOXX(1,1),BOXY(1,1))
      CALL VCHAR(BOXX(1,1)+36,BOXY(1,1)+60,'NEXT',4)
      CALL VCHAR(BOXX(1,1)+12,BOXY(1,1)+36,'QUESTION',8)
C **** DRAW THE 'PREVIOUS' BOX ****
      CALL VJLINE(4,BOXX(1,2),BOXY(1,2))
      CALL VCHAR(BOXX(1,2)+12,BOXY(1,2)+60,'PREVIOUS',8)
      CALL VCHAR(BOXX(1,2)+12,BOXY(1,2)+36,'QUESTION',8)
C **** DRAW THE 'ANSWER' BOX ****
      CALL VJLINE(4,BOXX(1,3),BOXY(1,3))
      CALL VCHAR(BOXX(1,3)+12,BOXY(1,3)+60,'QUESTION',8)
      CALL VCHAR(BOXX(1,3)+24,BOXY(1,3)+36,'ANSWER',6)
C **** DRAW THE 'SELECT' BOX ****
      CALL VJLINE(4,BOXX(1,4),BOXY(1,4))
      CALL VJLINE(4,BOXX(1,5),BOXY(1,5))
      CALL VPOSIT(MARK(2))
C **** GET THE QUESTIONS AND CHOICES ****
  100 CALL VORG(MARK(2))
      CALL VPOSIT(ITERM)
      CALL VTRUNC(ITERM)
      DO 200 I=1,4
  200 CALL VCHAR(QPOSX,QPOSY-(I-1)*24,QUEST(1,I,INDEX),64)
      CALL VPOSIT(MARK(3))
      DO 300 I=1,3
  300 CALL VCHAR(R1X,R1Y-(I-1)*24,CHOICE(1,I,1,INDEX),32)
      CALL VPOSIT(MARK(4))
      DO 400 I=1,3
  400 CALL VCHAR(R2X,R2Y-(I-1)*24,CHOICE(1,I,2,INDEX),32)
      CALL VPOSIT(MARK(5))
      DO 500 I=1,3
  500 CALL VCHAR(R3X,R3Y-(I-1)*24,CHOICE(1,I,3,INDEX),32)
      CALL VPOSIT(MARK(6))
C **** DRAW RESPONSE DIAGRAM OUTLINE ****
  600 IF((RESPON(1,INDEX).EQ.0.).AND.(RESPON(2,INDEX).EQ.0.).AND.
     X (RESPON(3,INDEX).EQ.0.)) GO TO 1000
C **** IF NO RESPONSE SKIP TO DRAW TRIANGLE *****
      CALL VORG(MARK(6))
      CALL VLINES(1,12,718,312,718)
C **** DRAW THE RESPONSES ****
      DO 900 I=1,3
      SCR(I)=VALUE(RESPON(I,INDEX))
      J=-1
      IF(SCR(I).GE.0) J=1
      ISCR=IABS(SCR(I))
      II=ISCR/4
      IJ=MOD(ISCR,4)
      IF(II.EQ.0) GO TO 800
```

```
      DO 700 K=1,II
      CALL VCHAR(12+(I-1)*100,725+J*K*5,PTS(5),4)
  700 CONTINUE
  800 CALL VCHAR(12+(I-1)*100,725+J*(II+1)*5,PTS(IJ+1),4)
  900 CONTINUE
      CALL VCHAR(12,850,'  A ',4)
      CALL VCHAR(112,850,'  B ',4)
      CALL VCHAR(212,850,'  C ',4)
      CALL FORMAT(1,4,0,SCR(1),ITERM)
      CALL VCHAR(12,600,ITERM,4)
      CALL FORMAT(1,4,0,SCR(2),ITERM)
      CALL VCHAR(112,600,ITERM,4)
      CALL FORMAT(1,4,0,SCR(3),ITERM)
      CALL VCHAR(212,600,ITERM,4)
      CALL VPOSIT(MARK(7))
C **** DRAW THE TRIANGLE ****
 1000 CALL VJLINE(3,TRIX,TRIY)
      CALL VCHAR(TRIX(1),TRIY(1)+24,'A',1)
      CALL VCHAR(TRIX(2)-24,TRIY(2)-12,'B',1)
      CALL VCHAR(TRIX(3)+24,TRIY(3)-12,'C',1)
      UNIT=FLOAT(TRIX(3)-TRIX(2))*(SQRT(3.)/2.)
      D1=RESPON(1,INDEX)*UNIT
      D2=RESPON(2,INDEX)*UNIT
      D3=RESPON(3,INDEX)*UNIT
      IF((D1.EQ.0.).AND.(D2.EQ.0.).AND.(D3.EQ.0.)) GO TO 1100
      IX0=(D1*.5+D3)/.866+TRIX(2)
      IY0=D1+TRIY(2)
      CALL VCHAR(IX0,IY0,'X',1)
 1100 CALL VPOSIT(MARK(8))
C ***** SEND THE TEST PICTURE *****
 1200 CALL VORG(MARK(8))
      DO 1300 I=1,N
      IF((RESPON(1,I).EQ.0.).AND.(RESPON(2,I).EQ.0.).AND.
     X (RESPON(3,I).EQ.0.)) GO TO 1400
 1300 CONTINUE
      CALL VCHAR(BOXX(1,4)+12,BOXY(1,4)+60,'SCORE',5)
      CALL VCHAR(BOXX(1,4)+12,BOXY(1,4)+36,'PAGE',4)
      MODE=1
      GO TO 1410
 1400 CALL VCHAR(BOXX(1,4)+36,BOXY(1,4)+60,'TEST',4)
      CALL VCHAR(BOXX(1,4)+24,BOXY(1,4)+36,'SELECT',6)
      MODE=0
 1410 CALL VPOSIT(MARK(9))
      CALL VTRUNC(MARK(9))
      CALL VSEND
C **** WAIT FOR THE LIGHT PEN ****
 1500 CALL PEN(IDATA,BOXX,BOXY,NARRAY,NBOX)
      GO TO (1800,2100,2200,2600,2800), NBOX
C **** PROCESS THE TRIANGLE ****
C ***** SET UP TRIANGULAR UNIT ********
```

```
      UNIT=TRIX(3)-TRIX(2)
      X=FLOAT(IDATA(1)-TRIX(2))/UNIT
      Y=FLOAT(IDATA(2)-TRIY(2))/UNIT
      D1=(SQRT(3.)*Y)/3.
      D2=1.-D1-X
      D3=X-D1
      D1=D1*2.
      IF(D1.LT.(-.025)) GO TO 1500
      IF(D2.LT.(-.025)) GO TO 1500
      IF(D3.LT.(-.025)) GO TO 1500
      IF(LOCKED(INDEX).EQ.0) GO TO 1600
      CALL VORG(MARK(9))
      CALL VCHAR(TRIX(2)+48,TRIY(2)+24,THEANS,36)
      CALL VPOSIT(ITERM)
      CALL VTRUNC(ITERM)
      CALL VSEND
      GO TO 1500
 1600 IF(D1.LT.(.025)) D1=0.
      IF(D2.LT.(.025)) D2=0.
      IF(D3.LT.(.025)) D3=0.
      TSUM=D1+D2+D3
      D1=D1/TSUM
      D2=D2/TSUM
      D3=D3/TSUM
      IF(    (ABS(D1-.3333).GT..05)
     X .OR. (ABS(D2-.3333).GT..05)
     X .OR. (ABS(D3-.3333).GT..05)) GO TO 1700
      D1=1./3.
      D2=D1
      D3=D1
 1700 RESPON(1,INDEX)=D1
      RESPON(2,INDEX)=D2
      RESPON(3,INDEX)=D3
      GO TO 600
C **** PROCESS 'NEXT' BOX ****
 1800 INDEX=INDEX+1
      IF(INDEX.LE.N) GO TO 100
      DO 1900 I=1,N
      IF((RESPON(1,I).EQ.0.).AND.(RESPON(2,I).EQ.0.).AND.
     X (RESPON(3,I).EQ.0.)) GO TO 2000
 1900 CONTINUE
 2000 INDEX=I
      GO TO 100
C **** PROCESS 'PREVIOUS' BOX ****
 2100 IF(INDEX.EQ.1) GO TO 1500
      INDEX=INDEX-1
      GO TO 100
C ***** PROCESS THE 'ANSWER' BOX *****
 2200 CONTINUE
      IF((RESPON(1,INDEX).EQ.0.).AND.(RESPON(2,INDEX).EQ.0.).AND.
```

```
X (RESPON(3,INDEX).EQ.O.)) GO TO 1500
  LOCKED(INDEX)=1
  CALL VORG(MARK(7))
  IX(1)=(ANSW(INDEX)-1)*100
  IX(2)=IX(1)+70
  IX(3)=IX(2)
  IX(4)=IX(1)
  IX(5)=IX(1)
  IY(1)=850-20
  IY(2)=IY(1)
  IY(3)=IY(1)-250
  IY(4)=IY(3)
  IY(5)=IY(1)
  CALL VJLINE(4,IX,IY)
  POTU=FLOAT(MAXPLT)*ABS(VALUE(1.))
  POTL=FLOAT(MAXPLT)*ABS(VALUE(0.))
  TOTAL=POTU+POTL
  IORGX=0
  IORGY=0
  IXUNIT=725
  AXUNIT=IXUNIT
  IYUNIT=525
  AYUNIT=IYUNIT
  IX(1)=IORGX+IXUNIT
  IX(2)=IORGX
  IX(3)=IORGX
  IX(4)=IX(1)
  IX(5)=IX(1)
  IX(6)=IORGX
  IX(7)=IORGX
  IX(8)=IX(1)
  IY(1)=IORGY
  IY(2)=IORGY+IFIX(POTL*AYUNIT/TOTAL)
  IY(3)=IORGY
  IY(4)=IORGY
  IY(5)=IORGY+IYUNIT
  IY(6)=IY(5)
  IY(7)=IY(2)
  IY(8)=IY(5)
  CALL VLTYPE(2)
  CALL VJLINE(7,IX,IY)
  CALL VLTYPE(1)
  DY1=FLOAT((IY(5)-IY(2))*14)/AXUNIT
  DY2=FLOAT((IY(2)-IY(1))*14)/AXUNIT
  DO 2300 J=1,14
  AJ=FLOAT(J)
  IDY1=AJ*DY1
  IDY2=AJ*DY2
  CALL VCHAR(IORGX+14*J,IY(2)+14+IDY1,MMAX(J),1)
  CALL VCHAR(IORGX+14*J,IY(2)-14-IDY2,MMIN(J),1)
```

```
2300 CONTINUE
     IX(1)=IORGX
     IY(1)=IORGY+IFIX(POTL*AYUNIT/TOTAL)
     NPLOT=0
     DO 2400 J=1,N
     IF((RESPON(1,J).EQ.0.).AND.(RESPON(2,J).EQ.0.).AND.
    X (RESPON(3,J).EQ.0.)) GO TO 2400
     NPLOT=NPLOT+1
     ZZ=VALUE(RESPON(ANSW(J),J))
     POTU=POTU-ZZ
     POTL=POTL+ZZ
     IX(NPLOT+1)=IORGX+(NPLOT*IXUNIT)/MAXPLT
     IY(NPLOT+1)=IORGY+IFIX(POTL*AYUNIT/TOTAL)
2400 CONTINUE
     CALL VJLINE(NPLOT,IX,IY)
     IX(1)=IX(NPLOT+1)+12
     IY(1)=IY(NPLOT+1)
     IX(2)=IX(1)+24
     IY(2)=IY(1)
     CALL VCHAR(IX(1),IY(1),'+',1)
     CALL VJLINE(1,IX,IY)
     CALL VCHAR(IX(2)+24,IY(1)+12,'YOUR',4)
     CALL VCHAR(IX(2)+24,IY(1)-12,'POINTS',6)
     IF (NPLOT.NE.N) GO TO 2500
     IX(1)=IX(NPLOT+1)
     IX(2)=IX(1)
     IY(1)=IORGY
     IY(2)=IY(1)+IYUNIT
     CALL VJLINE(1,IX,IY)
     CALL VCHAR(IX(1)-8,IY(2)+12,'END',3)
2500 CALL VPOSIT(MARK(8))
     GO TO 1200
C ***** PROCESS THE 'SELECT' BOX *****
2600 IF(MODE.EQ.0) GO TO  2700
C ************** GO TO SCORES ********************
     NEXT=3
     RETURN
C ************** GO TO TEST SELECT ***************
2700 NEXT=1
     RETURN
C ***** PRODUCE HARDCOPY **************
2800 CALL VCOPY(2)
     GO TO 1500
     END
```

SCORING SUBROUTINE

```
      SUBROUTINE PRINT(/MDIM/,/INDEX/,/N/,/RESPON/,/ANSW/,/MARK/,
     X               /BLANKS/,/TEST/,/SUBJCT/,/GEOM/,/PVAR/,/SCORE/,
     X               /SREAL/,/TREAL/,/SBASE/,/TBASE/,/NEXT/)
      DIMENSION RESPON(3,MDIM),MARK(1),TEST(1),SUBJCT(1),BLANKS(1)
      DIMENSION ANSW(MDIM)
C *********** DIMENSION STATEMENTS ******************
      DIMENSION PAR1(11),PAR2(33),PAR3(13)
      DIMENSION PAR4(16),PAR5(31)
      DIMENSION IDATA(2)
      DIMENSION SC(3),LOC(4)
      DIMENSION BOXX(5,5),BOXY(5,5)
      DIMENSION XFRO(7),XTO(7),YFRO(7),YTO(7)
      DIMENSION MUCHM(3),MUCHL(3)
      DIMENSION SLEEP(11),FOOL(11)
      DIMENSION CONS(3),SOME(2),OVER(3),UNDER(3),KNOW(4)
      DIMENSION RES1(10),RES2(4),RES3(2),RES4(12)
      DIMENSION TEMPR(3),CHAR(3),ASTER(3)
C *********** TYPE STATEMENTS *********************
      INTEGER SLEEP,FOOL,ASTER
      INTEGER ANSW,MARK,BLANKS,TEST,SUBJCT,SCORE,ISCORE
      INTEGER PAR2,PAR3
      INTEGER PAR4,PAR5
      INTEGER RES1,RES2,RES3,RES4
      INTEGER XFRO,XTO,YFRO,YTO
      INTEGER MUCHM,MUCH,MUCHL,LESS
      INTEGER EASY,HARD
      INTEGER CONS,SOME,OVER,UNDER,KNOW
      INTEGER IDATAY,NARRAY
      INTEGER COORXF,COORYT
      INTEGER SC,CHAR,XSCORE
      INTEGER BOXX,BOXY
      REAL HMAX
C *********** DATA STATEMENTS **********************
      DATA XORG/300./,YORG/500./,XUNIT/400./,YUNIT/400./
      DATA COORXF/50/,COORYT/450/
      DATA SC/'SCOR','E=  ','    '/
      DATA CHAR/'A   ','B   ','C   '/
      DATA SLEEP/'U   ','N   ','D   ','E   ','R   ','-   ',
     X          'V   ','A   ','L   ','U   ','E   '/
      DATA FOOL /'O   ','V   ','E   ','R   ','-   ',
     X          'V   ','A   ','L   ','U   ','E   ','    '/
      DATA  BOXX/875,975,975,875,875,
     X           875,975,975,875,875,
     X           875,975,975,875,875,
     X           875,975,975,875,875,
```

```
X                0,10,10,0,0/
   DATA   BOXY/10,10,85,85,10,
X               110,110,185,185,110,
X               210,210,285,285,210,
X               310,310,385,385,310,
X               990,990,999,999,990/
   DATA EASY/'EASY'/,HARD/'HARD'/
   DATA MUCHM /' MUC','H MO','RE  '/
   DATA MORE  /'MORE'/
   DATA MUCHL /' MUC','H LE','SS  '/
   DATA LESS  /'LESS'/
   DATA SOME/'SOME','WHAT'/
   DATA CONS/'CONS','IDER','ABLY'/
   DATA OVER/' OVE','R VA','LUE '/
   DATA UNDER/' UND','ER V','ALUE'/
   DATA KNOW/' YOU','R KN','OWLE','DGE.'/
   DATA RES1/'BY M','ORE ','REAL','ISTI','C US','E OF',
X               'YOU','R KN','OWLE','DGE.'/
   DATA RES2/'BY M','ORE ','STUD','Y.  '/
   DATA RES3/'OVER','ALL.'/
   DATA RES4/' HEA','VILY',' ON ','ANSW','ERS ','YOU ',
X             'FEEL',' ARE',' COR','RECT','.   ','    '/
   DATA PAR1/'YOU ','CAN ','IMPR','OVE ','YOUR',' SCO',
X             'RE  ','BY  ','    ','POI ','NTS '/
   DATA PAR2 /'THIS',' TES','T  W','AS  ','SO  ',
X             '    ','FOR ',' YOU','THA','T IT',' IS ',
X             'IMPO','SSIB','LE T','O DE','TERM',
X             'INE ','HOW ','DISC','RIMI','NATI','NG  ',
X             'YOU ','ARE ','AT V','ALUI','NG Y',
X             'OUR ','FACT','S AN','D RE','ASON','S.  '/
   DATA PAR3 /'YOU ','TEND',' TO ',10*' '/
   DATA PAR4 /'BET ',15*' '/
   DATA PAR5 /'THE ','WAY ','YOU ','PLAC','ED Y',
X             'OUR ','BETS',' ON ','THIS',' TES','T   ',
X             'SHOW','S YO','U AR','E VE','RY A',
X             'CCUR','ATE ','IN A','SSES','SING',
X             'THE ','VALI','DITY',' OF ','YOUR',
X             ' FAC','TS A','ND R','EASO','NS. '/
   DATA ISKS/'**  '/
   DATA NARRAY/5/
C ****** FORMAT*******************************************
 5000 FORMAT('1','TEST:',5A4,5X,'SUBJECT:',16A4)
 5100 FORMAT('0','DATE:',2A4,5X,'TIME:',2A4)
 5200 FORMAT('0','SCORE:',I5,5X,'REVISED SCORE:',I5,
X          5X,'STEREOTYPE SCORE:',I5)
 5300 FORMAT('0','INFORMATION:',F6.2,8X,'REVISED INFORMATION:',F6.2)
 5400 FORMAT('0','REALISM SLOPE:',F7.2,5X,'REALISM INTERCEPT:',
X          F7.2)
 5500 FORMAT('0',6X,'PROBABILITIES',23X,'SCORES')
 5600 FORMAT('  ',3X,'A',7X,'B',7X,'C',13X,'A',10X,'B',10X,'C')
```

```
 5700 FORMAT(' ',F6.2,A2,F6.2,A2,F6.2,A2,
      X          3X,F9.2,A2,F9.2,A2,F9.2,A2)
 5800 FORMAT('0','BASE SLOPE:',F7.2,8X,'BASE INTERCEPT:',
      X          F7.2)
 5900 FORMAT('0','BIAS:',F5.2,16X,'PROB. VARIANCE:',F6.2)
C ********* THE SCORING FUNCTION ************************
      VALUE(X)=50.*ALOG10(3.*AMAX1(.01,X))
C ******** DO THE REALISM BIT **************************
      CALL REAL(MDIM,N,RESPON,ANSW,SCORE,GEOM,
      X          SREAL,TREAL,SBASE,TBASE,PVAR)
C ******** DISPLAY TO INFORMATION BOX ******************
      XFRO(1)=XORG
      XTO(1)=XORG+XUNIT
      YFRO(1)=YORG
      YTO(1)=YORG
      XFRO(2)=XTO(1)
      XTO(2)=XTO(1)
      YFRO(2)=YORG
      YTO(2)=YORG+YUNIT
      XFRO(3)=XTO(1)
      XTO(3)=XORG
      YFRO(3)=YTO(2)
      YTO(3)=YTO(2)
      XFRO(4)=XORG
      XTO(4)=XORG
      YFRO(4)=YTO(2)
      YTO(4)=YORG
      XFRO(5)=XORG
      XTO(5)=XTO(1)
      YFRO(5)=YORG
      YTO(5)=YTO(2)
      XFRO(6)=XORG
      XTO(6)=XTO(1)
      YFRO(6)=YTO(2)
      YTO(6)=YORG
      SUMA=0.
      SUMP=0.
      HMAX=ALOG10(3.)*FLOAT(N)
      AI=0.
      PI=0.
      IF (PVAR.LT..01) GO TO 700
      DO 600 I=1,N
      DO 100 J=1,3
      TEMP=RESPON(J,I)
      IF(TEMP.GT..0001) SUMP=SUMP+TEMP*ALOG10(TEMP)
  100 CONTINUE
      K=1
      DO 300 J=1,3
      TEMP=SREAL*RESPON(J,I)+TREAL
      IF(TEMP.LT.0.) TEMP=0.
```

```
      IF(TEMP.GT.1.) TEMP=1.
      IF(J.NE.ANSW(I)) GO TO 200
      TEMPR(1)=TEMP
      GO TO 300
  200 K=K+1
      TEMPR(K)=TEMP
  300 CONTINUE
      DIFF=1.-TEMPR(1)
      IF((TEMPR(2).NE.0.).OR.(TEMPR(3).NE.0.)) GO TO 400
      TEMPR(2)=.5
      TEMPR(3)=.5
  400 DENOM=TEMPR(2)+TEMPR(3)
      TEMPR(2)=TEMPR(2)*DIFF/DENOM
      TEMPR(3)=TEMPR(3)*DIFF/DENOM
      DO 500 J=1,3
  500 IF(TEMPR(J).GT..0001) SUMA=SUMA+TEMPR(J)*ALOG10(TEMPR(J))
  600 CONTINUE
      PI=HMAX+SUMP
      AI=HMAX+SUMA
  700 CONTINUE
      XFRO(7)=XORG
      XTO(7)=XTO(1)
      YFRO(7)=YORG+(AI/HMAX)*YUNIT
      YTO(7)=YORG+(PI/HMAX)*YUNIT
      CALL VORG(MARK(1))
      CALL VLTYPE(2)
      CALL VLINES(6,XFRO,YFRO,XTO,YTO)
      CALL VLTYPE(1)
      CALL VLINES(1,XFRO(7),YFRO(7),XTO(7),YTO(7))
      CALL VCHAR(XFRO(7)-160,YFRO(7)+24,'ACTUAL',6)
      CALL VCHAR(XFRO(7)-160,YFRO(7),'KNOWLEDGE-',10)
      CALL VCHAR(XTO(7)+12,YTO(7)+24,'PERCEIVED',9)
      CALL VCHAR(XTO(7),YTO(7),'-KNOWLEDGE',10)
      CALL VCHAR(XFRO(1)+24,YFRO(3)+12,
     X            'COMPLETE MASTERY',16)
      CALL VCHAR(XFRO(1)+24,YFRO(1)-12,
     X            'NO MASTERY',10)
      IT1X=XFRO(1)+24
      IT1Y=YFRO(3)-48
      IT2X=XFRO(1)+24
      IT2Y=YFRO(1)+48
      DO 800 I=1,11
      CALL VCHAR(IT1X+(I-1)*12,IT1Y-(I-1)*12,SLEEP(I),1)
      CALL VCHAR(IT1X+(I-1)*12,IT2Y+(I-1)*12,FOOL(I),1)
  800 CONTINUE
C ************** DISPLAY VERBAL RESPONSE *************
      BYAS=(PI-AI)/HMAX
      CALL DATIME(2,LOC)
      ITERM=YORG+YUNIT+75.
      CALL VCHAR(COORXF,ITERM,TEST,20)
```

```
          CALL VCHAR(COORXF+240,ITERM,LOC(1),8)
          CALL VCHAR(COORXF+360,ITERM,LOC(3),8)
          CALL VCHAR(COORXF+480,ITERM,SUBJCT,20)
C ***** ISSUE THE MESSAGES *****
C *****   TOO HARD? *****
          IF(PVAR.GT..01) GO TO  900
          PAR2(6)=HARD
          CALL VCHAR(COORXF,COORYT-5*24,PAR2(1),44)
          CALL VCHAR(COORXF,COORYT-6*24,PAR2(12),44)
          CALL VCHAR(COORXF,COORYT-7*24,PAR2(23),44)
          GO TO 3200
C ***** REALLY GOOD? *****
  900 IF(ABS(BYAS).GT.(.1)) GO TO 1000
          CALL VCHAR(COORXF,COORYT-5*24,PAR5(1),44)
          CALL VCHAR(COORXF,COORYT-6*24,PAR5(12),40)
          CALL VCHAR(COORXF,COORYT-7*24,PAR5(22),40)
          GO TO 3000
 1000 J=0
          JJ=1
          DO 1100 K=1,10
 1100 PAR3(3+K)=BLANKS(1)
          DO 1200 K=1,15
 1200 PAR4(1+K)=BLANKS(1)
C ***** OVER-VALUING? *****
          IF(BYAS.LT.(-.1)) GO TO 2100
          IF(BYAS.LE.(.7)) GO TO 1400
          DO 1300 J=1,3
 1300 PAR3(3+J)=CONS(J)
          PAR4(2)=MUCHL(1)
          PAR4(3)=MUCHL(2)
          PAR4(4)=MUCHL(3)
          JJ=3
          GO TO 1700
 1400 IF(BYAS.GE.(.4)) GO TO 1600
          DO 1500 J=1,2
 1500 PAR3(3+J)=SOME(J)
          PAR4(2)=LESS
          GO TO 1700
 1600 PAR4(2)=LESS
 1700 DO 1800 K=1,3
 1800 PAR3(3+J+K)=OVER(K)
          DO 1900 L=1,4
 1900 PAR3(3+J+K+L)=KNOW(L)
          DO 2000 K=1,12
 2000 PAR4(1+JJ+K)=RES4(K)
          CALL VCHAR(COORXF,COORYT-5*24,PAR3(1),52)
          CALL VCHAR(COORXF,COORYT-8*24,PAR4(1),64)
          GO TO 3000
C ***** UNDER-VALUING? *****
 2100 IF(BYAS.GT.(-.7)) GO TO 2300
```

```
      DO 2200 J=1,3
 2200 PAR3(3+J)=CONS(J)
      PAR4(2)=MUCHM(1)
      PAR4(3)=MUCHM(2)
      PAR4(4)=MUCHM(3)
      JJ=3
      GO TO 2600
 2300 IF(BYAS.LT.(-.4)) GO TO 2500
      DO 2400 J=1,2
 2400 PAR3(3+J)=SOME(J)
      PAR4(2)=MORE
      GO TO 2600
 2500 PAR4(2)=MORE
 2600 DO 2700 K=1,3
 2700 PAR3(3+J+K)=UNDER(K)
      DO 2800 L=1,4
 2800 PAR3(3+J+K+L)=KNOW(L)
      DO 2900 K=1,12
 2900 PAR4(1+JJ+K)=RES4(K)
      CALL VCHAR(COORXF,COORYT-5*24,PAR3(1),52)
      CALL VCHAR(COORXF,COORYT-8*24,PAR4(1),64)
C ******* PRINT THE SCORES *********************
 3000 CONTINUE
      RSCORE=0.
      DO 3100 K=1,N
      TEMP=(SREAL*RESPON(ANSW(K),K)+TREAL)
      IF(TEMP.GT.1.) TEMP=1.
      IF(TEMP.LT..01) TEMP=.01
      RSCORE=RSCORE+VALUE(TEMP)
 3100 CONTINUE
      IF((RSCORE-SCORE).LT.0.) RSCORE=SCORE
      ITERM=RSCORE-SCORE+.5
      CALL FORMAT(1,4,0,ITERM,PAR1(9))
      CALL VCHAR(COORXF,COORYT-11*24,PAR1,44)
      CALL VCHAR(COORXF,COORYT-12*24,RES1,40)
      ITERM=FLOAT(N*24)-RSCORE+.5
      CALL FORMAT(1,4,0,ITERM,PAR1(9))
      CALL VCHAR(COORXF,COORYT-14*24,PAR1,44)
      CALL VCHAR(COORXF,COORYT-15*24,RES2,16)
      ITERM=FLOAT(N*24)-SCORE+.5
      CALL FORMAT(1,4,0,ITERM,PAR1(9))
      CALL VCHAR(COORXF,COORYT-17*24,PAR1,44)
      CALL VCHAR(COORXF,COORYT-18*24,RES3,8)
C **** DRAW THE 'TEST AND PLOT' BOXES ****
 3200 CALL VPOSIT(MARK(2))
 3300 CALL VORG(MARK(2))
      CALL VJLINE(4,BOXX(1,1),BOXY(1,1))
      CALL VCHAR(BOXX(1,1)+12,BOXY(1,1)+60,'OLD',3)
      CALL VCHAR(BOXX(1,1)+12,BOXY(1,1)+36,'TEST ',6)
      CALL VJLINE(4,BOXX(1,2),BOXY(1,2))
```

```
      CALL VCHAR(BOXX(1,2)+12,BOXY(1,2)+36,'PLOT  ',6)
      CALL VJLINE(4,BOXX(1,3),BOXY(1,3))
      CALL VCHAR(BOXX(1,3)+12,BOXY(1,3)+36,'RECORD',6)
      CALL VJLINE(4,BOXX(1,4),BOXY(1,4))
      CALL VCHAR(BOXX(1,4)+12,BOXY(1,4)+60,'NEW ',4)
      CALL VCHAR(BOXX(1,4)+12,BOXY(1,4)+36,'TEST',4)
      CALL VJLINE(4,BOXX(1,5),BOXY(1,5))
      CALL VPOSIT(ITERM)
      CALL VTRUNC(ITERM)
      CALL VSEND
C ***** ***** ***** ***** ***** ***** ***** *****
C ***** WAIT FOR THE LIGHT PEN *****************
 3400 CALL PEN(IDATA,BOXX,BOXY,NARRAY,NBOX)
      GO TO (3500,3600,3700,4200,4300), NBOX
      GO TO 3400
C ************ RETURN TO OLD TEST *********************
 3500 NEXT=2
      RETURN
C ************ GO TO PLOT ROUTINE *********************
 3600 NEXT=4
      RETURN
C ************ PRINT HARDCOPY *************************
 3700 CALL VORG(MARK(2))
      CALL VTRUNC(MARK(2))
      CALL VSEND
      TEMP=0.
      PERC1=(1.-SBASE-TBASE)/2.
      PERC2=SBASE+TBASE
      DO 3900 I=1,N
      PERC=PERC1
      E=1.
      TEM=AMAX1(RESPON(1,I),RESPON(2,I),RESPON(3,I))
      IF(ABS(RESPON(ANSW(I),I)-TEM).GT..01) GO TO 3900
      PERC=PERC2
      E=0.
      DO 3800 J=1,3
 3800 IF(ABS(RESPON(J,I)-TEM).LE..01) E=E+1.
 3900 TEMP=TEMP+VALUE(PERC)/E
      XSCORE=TEMP
      PRINT 5000,(TEST(J),J=1,5),(SUBJCT(J),J=1,16)
      PRINT 5100,(LOC(J),J=1,2),(LOC(J),J=3,4)
      ISCORE=RSCORE
      PRINT 5200,SCORE,ISCORE,XSCORE
      TEMP1=PI/HMAX
      TEMP2=AI/HMAX
      PRINT 5300,TEMP1,TEMP2
      PRINT 5400,SREAL,TREAL
      PRINT 5800,SBASE,TBASE
      PRINT 5900,BYAS,PVAR
      PRINT 5500
```

```
      PRINT 5600
      DO 4100 I=1,N
      DO 4000 J=1,3
 4000 TEMPR(J)=VALUE(RESPON(J,I))
      ASTER(1)=BLANKS(1)
      ASTER(2)=BLANKS(2)
      ASTER(3)=BLANKS(3)
      ASTER(ANSW(I))=ISKS
      PRINT 5700,RESPON(1,I),ASTER(1),RESPON(2,I),ASTER(2),
    X  RESPON(3,I),ASTER(3),TEMPR(1),ASTER(1),TEMPR(2),
    X  ASTER(2),TEMPR(3),ASTER(3)
 4100 CONTINUE
      GO TO 3300
C ********* GET A NEW TEST ******************************
 4200 NEXT=1
      RETURN
C ********* PLOT HARDCOPY ***************************
 4300 CALL VCOPY(2)
      GO TO 3400
      END
```

EXTERNAL VALIDITY PLOT SUBROUTINE

```
      SUBROUTINE PLOT(/MDIM/,/MARK/,/SCORE/,/GEOM/,/TEST/,
     X              /SUBJCT/,/SREAL/,/TREAL/,/SBASE/,/TBASE/,
     X              /N/,/RESPON/,/ANSW/,/NEXT/)
      DIMENSION MARK(1),TEST(1),SUBJCT(1),RESPON(3,MDIM),ANSW(1)
C ********* DIMENSION STATEMENTS ********************************
      DIMENSION COORXF(5),COORXT(5),COORYT(5),COORYF(5)
      DIMENSION SC(4),XSC(4),TICS(11)
      DIMENSION HISTXF(10),HISTXT(10),HISTYF(10),HISTYT (10)
      DIMENSION PLOTXF(1),PLOTXT(1),PLOTYT(1),PLOTYF(1)
      DIMENSION BOXX(5,3),BOXY(5,3)
      DIMENSION IDATA(2),LOC(4)
C ********* TYPE DECLARATIONS **********************************
      INTEGER MARK,TEST,SUBJCT,SCORE,XSCORE,ANSW
      INTEGER COORXT,COORXF,COORYT,COORYF
      INTEGER SC,XSC
      INTEGER HISTXF,HISTXT,HISTYF,HISTYT
      INTEGER PLOTXT,PLOTXF,PLOTYT,PLOTYF
      INTEGER BOXX,BOXY
      INTEGER IDATA,NARRAY
C ************ DATA STATEMENTS *********************************
      DATA COORXF/200,200,200,200,200/
      DATA COORXT/800,200,800,200,800/
      DATA COORYF/200,200,200,200,200/
      DATA COORYT/200,800,200,800,800/
      DATA SC/'SCOR','E=    ','      ','      '/
      DATA XSC/'XSCO','RE=   ','      ','      '/
      DATA TICS/'  0 ','  .1','  .2','  .3','  .4','  .5'
     X ,'  .6','  .7','  .8','  .9','1.0 '/
      DATA BOXX/875,975,975,875,875,
     X          875,975,975,875,875,
     X          0,10,10,0,0/
      DATA BOXY/10,10,85,85,10,
     X          110,110,185,185,110,
     X          990,990,999,999,990/
      DATA NARRAY/3/
C ******* THE SCORING FUNCTION *****************************
      VALUE(X)=50.*ALOG10(3.*AMAX1(.01,X))
C ******** COMPUTE THE STEREOTYPE SCORE ************
      TEMP=0.
      PERC1=(1.-SBASE-TBASE)/2.
      PERC2=SBASE+TBASE
      DO 200 I=1,N
      PERC=PERC1
      E=1.
      TEM=AMAX1(RESPON(1,I),RESPON(2,I),RESPON(3,I))
```

```
      IF(ABS(RESPON(ANSW(I),I)-TEM).GT..01) GO TO 200
      PERC=PERC2
      E=0.
      DO 100 J=1,3
  100 IF(ABS(RESPON(J,I)-TEM).LE..01) E=E+1.
  200 TEMP=TEMP+VALUE(PERC)/E
      XSCORE=TEMP
C ***** PLOT THE COORDINATE SYSTEM ****
      CALL VORG(MARK(1))
      CALL FORMAT(1,8,0,SCORE,SC(3))
      CALL FORMAT(1,8,0,XSCORE,XSC(3))
      CALL VCHAR(COORXF(1),COORYT(2)+50,SC,16)
      CALL VCHAR(COORXF(1)+300,COORYT(2)+50,XSC,16)
      CALL DATIME(2,LOC)
      CALL VCHAR(COORXF(1),COORYT(2)+100,TEST,20)
      CALL VCHAR(COORXF(1)+240,COORYT(2)+100,LOC(1),8)
      CALL VCHAR(COORXF(1)+360,COORYT(2)+100,LOC(3),8)
      CALL VCHAR(COORXF(1)+480,COORYT(2)+100,SUBJCT,20)
      CALL VLINES(5,COORXF,COORYF,COORXT,COORYT)
      ISTEPX=(COORXT(1)-COORXF(1))/10
      ISTEPY=(COORYT(2)-COORYF(2))/10
      DO  300 J=1,11
      MIX=COORXF(1)+(J-1)*ISTEPX
      MIY=COORYF(2)+(J-1)*ISTEPY
      CALL VCHAR(MIX,COORYF(1),'1',1)
      CALL VCHAR(MIX-12,COORYF(1)-36,TICS(J),4)
      CALL VCHAR(COORXF(2),MIY,'-',1)
      CALL VCHAR(COORXF(2)-60,MIY,TICS(J),4)
  300 CONTINUE
C ***** PLOT THE PROBABILITY DISTRIBUTION *****
      DO  400 J=1,10
      HISTYF(J)=0
      HISTYT(J)=0
  400 CONTINUE
      DO  500 I=1,N
      DO  500 J=1,3
      K=RESPON(J,I)*9.99+1.
      HISTYT(K)=HISTYT(K)+1
      IF(ANSW(I).EQ.J) HISTYF(K)=HISTYF(K)+1
  500 CONTINUE
      ISTEPY=(COORYT(2)-COORYF(2))
      DO  700 J=1,10
      HISTXF(J)=COORXF(1)+ISTEPX/2+(J-1)*ISTEPX
      HISTXT(J)=HISTXF(J)
      IF(HISTYT(J).EQ.0) GO TO 600
      ITEMP=COORYF(2)+(HISTYF(J)*ISTEPY)/HISTYT(J)
      CALL VCHAR(HISTXF(J),ITEMP,'X',1)
  600 HISTYT(J)=COORYF(2)+(HISTYT(J)*ISTEPY)/(3*N)
      HISTYF(J)=COORYF(2)
  700 CONTINUE
```

```
      CALL VLINES(10,HISTXF,HISTYF,HISTXT,HISTYT)
C ***** PLOT THE REALISM LINES ****
      UNIT=FLOAT(COORXT(1)-COORXF(1))
      YORG=FLOAT(COORYF(1))
      XORG=FLOAT(COORXF(1))
      IF((TREAL.GE.0.).AND.(TREAL.LE.1.)) TY1=TREAL
      IF((SREAL.LT.0.).AND.(TREAL.GT.1.)) TY1=1.
      IF((SREAL.GT.0.).AND.(TREAL.LT.0.)) TY1=0.
      TEM=TREAL+SREAL
      IF((TEM.GE.0.).AND.(TEM.LE.1.)) TY2=TEM
      IF((SREAL.LT.0.).AND.(TEM.LT.0.)) TY2=0.
      IF((SREAL.GT.0.).AND.(TEM.GT.1.)) TY2=1.
      TX1=(TY1-TREAL)/SREAL
      TX2=(TY2-TREAL)/SREAL
      IF(ABS(SREAL).LT.(.01)) TX1=0.
      IF(ABS(SREAL).LT.(.01)) TX2=1.
      PLOTXF(1)=TX1*UNIT+XORG
      PLOTXT(1)=TX2*UNIT+XORG
      PLOTYF(1)=TY1*UNIT+YORG
      PLOTYT(1)=TY2*UNIT+YORG
      CALL VLINES(1,PLOTXF,PLOTYF,PLOTXT,PLOTYT)
      CALL VLINES(1,PLOTXF,PLOTYF,PLOTXT,PLOTYT)
      IF((TBASE.GE.0.).AND.(TBASE.LE.1.)) TY1=TBASE
      IF((SBASE.LT.0.).AND.(TBASE.GT.1.)) TY1=1.
      IF((SBASE.GT.0.).AND.(TBASE.LT.0.)) TY1=0.
      TEM=TBASE+SBASE
      IF((TEM.GE.0.).AND.(TEM.LE.1.)) TY2=TEM
      IF((SBASE.LT.0.).AND.(TEM.LT.0.)) TY2=0.
      IF((SBASE.GT.0.).AND.(TEM.GT.1.)) TY2=1.
      TX1=(TY1-TBASE)/SBASE
      TX2=(TY2-TBASE)/SBASE
      IF(ABS(SBASE).LT.(.01)) TX1=0.
      IF(ABS(SBASE).LT.(.01)) TX2=1.
      PLOTXF(1)=TX1*UNIT+XORG
      PLOTXT(1)=TX2*UNIT+XORG
      PLOTYF(1)=TY1*UNIT+YORG
      PLOTYT(1)=TY2*UNIT+YORG
      CALL VLTYPE(2)
      CALL VLINES(1,PLOTXF,PLOTYF,PLOTXT,PLOTYT)
      CALL VLTYPE(1)
C **** DRAW THE 'TEST AND PLOT' BOXES ****
      CALL VJLINE(4,BOXX(1,1),BOXY(1,1))
      CALL VCHAR(BOXX(1,1)+12,BOXY(1,1)+36,'TEST  ',6)
      CALL VJLINE(4,BOXX(1,2),BOXY(1,2))
      CALL VCHAR(BOXX(1,2)+12,BOXY(1,2)+60,'SCORE',5)
      CALL VCHAR(BOXX(1,2)+12,BOXY(1,2)+36,'PAGE',4)
      CALL VJLINE(4,BOXX(1,3),BOXY(1,3))
      CALL VPOSIT(ITERM)
      CALL VTRUNC(ITERM)
      CALL VSEND
```

```
C ***** ***** ***** ***** ***** ***** ***** *****
C ***** WAIT FOR THE LIGHT PEN ******************
  800 CALL PEN(IDATA,BOXX,BOXY,NARRAY,NBOX)
      GO TO (900,1000,1100), NBOX
      GO TO  800
  900 NEXT=2
      RETURN
 1000 NEXT=3
      RETURN
 1100 CALL VCOPY(2)
      GO TO 800
      END
```

REALISM PARAMETER SUBROUTINE

```
      SUBROUTINE REAL(/MDIM/,/N/,/RESPON/,/ANSW/,/SCORE/,
     X                /GEOM/,/SREAL/,/TREAL/,/SBASE/,/TBASE/,
     X                /PVAR/)
      DIMENSION RESPON(3,MDIM),ANSW(1)
C ************** TYPE DECLARATIONS *****************************
      INTEGER ANSW,SCORE
C ******** THE SCORING FUNCTION **************************
      VALUE(X)=50.*ALOG10(3.*AMAX1(.01,X))
C ******** COMPUTE THE REALISM VALUES ******************
      TSUME=0.
      TSUMK=0.
      TSUMPR=0.
      TSUMP2=0.
      DO  400 I=1,N
      E=0.
      DO  100 J=1,3
      TSUMP2=TSUMP2+RESPON(J,I)**2
  100 CONTINUE
      TEM=AMAX1(RESPON(1,I),RESPON(2,I),RESPON(3,I))
      IF(ABS(RESPON(ANSW(I),I)-TEM).GT.(.01)) GO TO  300
      DO  200 J=1,3
      IF(ABS(RESPON(J,I)-TEM).LE.(.01)) E=E+1.
  200 CONTINUE
  300 IF(E.NE.0.) TSUME=TSUME+1./E
      TSUMK=TSUMK+3.
      TSUMPR=TSUMPR+RESPON(ANSW(I),I)
  400 CONTINUE
      TSCORE=0.
      GEOM=1.
      DO  500 J=1,N
      TSCORE=TSCORE+VALUE(RESPON(ANSW(J),J))
      TEMP=RESPON(ANSW(J),J)
      IF(TEMP.LT.(.01)) TEMP=.01
      GEOM=GEOM*TEMP
  500 CONTINUE
      TN=FLOAT(N)
      GEOM=EXP(ALOG(GEOM)/TN)
      PBAR=TN/TSUMK
      DELTA=TSUMP2-TN*PBAR
      SREAL=(TSUMPR-PBAR*TN)/DELTA
      TREAL=PBAR*(1.-SREAL)
      SBASE=(TSUME/TN-PBAR)/(1.-PBAR)
      TBASE=PBAR*(1.-SBASE)
      SCORE=TSCORE
      PVAR=DELTA/TSUMK
      RETURN
      END
```

LIGHT PEN CONTROL SUBROUTINE

```
      SUBROUTINE PEN(/IDATA/,/BOXX/,/BOXY/,/NARRAY/,/NBOX/)
      DIMENSION IDATA(1),BOXX(5,NARRAY),BOXY(5,NARRAY)
      INTEGER BOXX,BOXY
C *********** WAIT FOR THE LIGHT BUTTON *************
      CALL VLPEN(2,1,1000,IDATA,ITERM)
      DO 100 J=1,NARRAY
      IF ((IDATA(1).GE.BOXX(1,J)).AND.
     X    (IDATA(1).LE.BOXX(2,J)).AND.
     X    (IDATA(2).GE.BOXY(2,J)).AND.
     X    (IDATA(2).LE.BOXY(3,J))) GO TO 200
  100 CONTINUE
      J=0
  200 NBOX=J
      RETURN
      END
```

REFERENCES

1. Shuford, E. H., Jr., A. Albert, and H. E. Massengill, "Admissible Probability Measurement Procedures," *Psychometrika*, 31, 1966, pp. 125-145.

2. Gardner, Willie C., *The Use of Confidence Testing in the Academic Instructor Course*, Proceedings of the 1969 Annual Meeting of the Military Testing Association, New York, N.Y., September 15-19, 1961.

3. Brown, Thomas A., *Probabilistic Forecasts and Reproducing Scoring Systems*, The Rand Corporation, RM-6299-ARPA, June 1970.

4. Uncapher, K., *The Rand Video Graphic System--An Approach to a General User-Computer Graphic Communication System*, The Rand Corporation, R-753-ARPA, April 1971.

5. Carlstedt, J., *A Message Exchange for Computer Programs and Terminals*, The Rand Corporation, R-694-ARPA, May 1972.

6. Clark, R. L., *VPACK: A FORTRAN-Callable Graphics Package for the Rand Video Graphic System*, The Rand Corporation, R-1015-NIH, December 1972.

7. Bitzer, D., and D. Skaperdas, *The Design of an Economically Viable Large-Scale, Computer-Based Education System*, University of Illinois CERL Report X-5, January 1972.

8. Stifle, J., *The PLATO IV Architecture*, University of Illinois CERL Report X-20, May 1972.

9. Avner, R. A., and Paul Tenczar, *The TUTOR Manual*, University of Illinois CERL Report X-4, 1969.